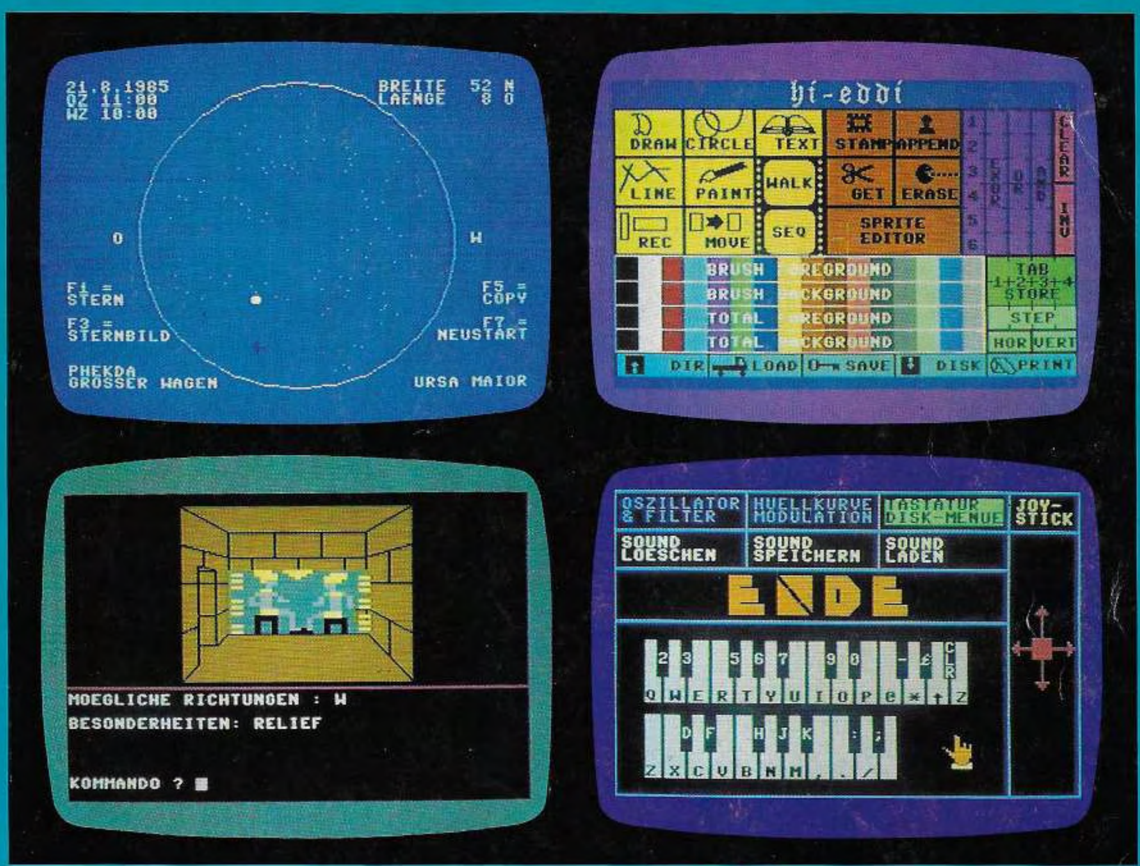




AUSGEWÄHLTE SUPER-LISTINGS

Grafik ★ Spiele ★ Anwendungen ★ Sport & Hobby ★ Floppy ★ Musik ★ Compiler ★ Alle Programme neu überarbeitet und mit vielen Ergänzungen

Alle Programme auch auf Diskette erhältlich



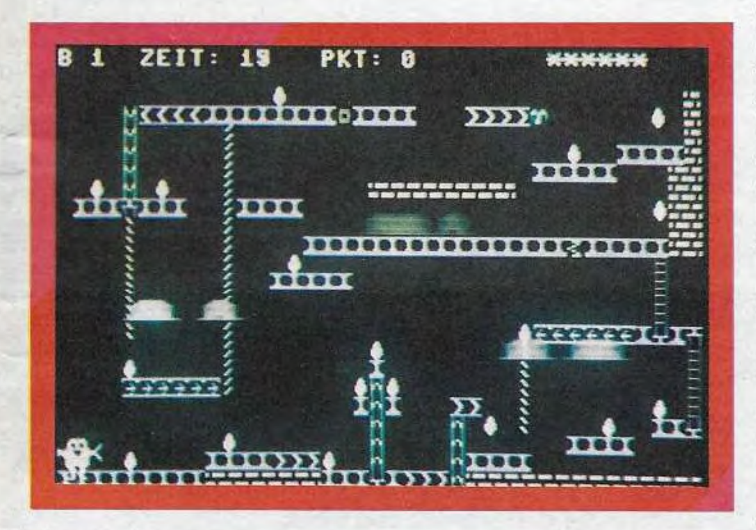
Das Beste vom Besten

Die besten Programme aus der 64'er: Ein Leckerbissen sondergleichen!

Im April 1984 wurde das erste Heft der 64'er, dem Magazin für Computerfans, geboren. Und schon seit dieser Ausgabe konnten wir Sie mit hervorragenden Programmen überraschen. Bereits im 64'er, Ausgabe 5/84 gab es das erste Listing des Monats, ein Zeichen für ein hervorragendes Programm, eine ausgefallene Idee, eben ein Spitzenprodukt. Es gab jedoch noch eine ganze Reihe von Listings, die dieses Prädikat auch verdient haben. Die besten von ihnen, darunter zwei Programme aus unserer Schwesterzeitschrift Happy-Computer, und eine Auswahl aller Listings des Monats haben wir für Sie in diesem Sonderheft zusammengestellt. Wahrlich ein Menü, das es in dieser Brisanz bisher nicht gegeben hat. Es gibt Programme zu den Themen

— Anwendung,
— Grafik,
— Hilfsprogramme (Utilities) und
— Spiele.

Alle Programme sind noch einmal überarbeitet worden. Wenn es Fehler gegeben hat, wurden sie verbessert. Auch Erweiterungen zu einigen Programmen, zum Beispiel zu HI-EDDI, haben wir nicht vergessen. Durch die beiden Eintipphilfen Checksummer (Seite 16) und MSE (Seite 18) werden Sie die Listings fehlerfrei eintippen können.

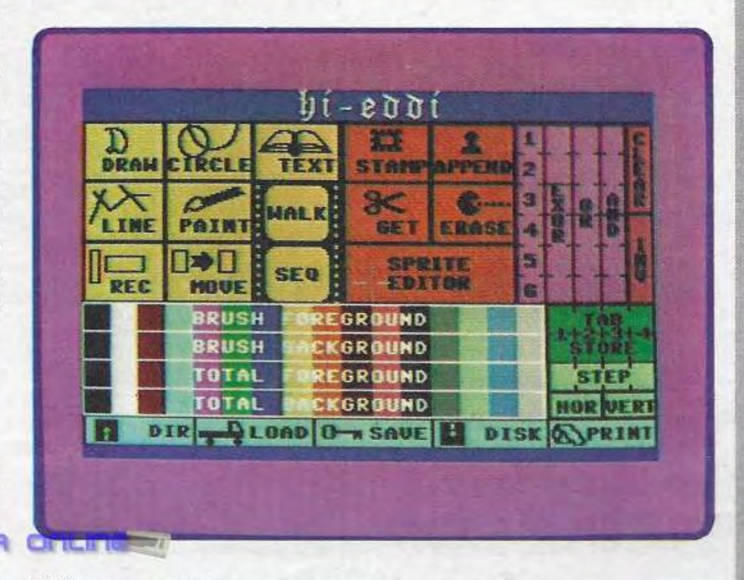


Anwendungen

Der C 64 ist bekannterweise nicht nur ein Computer zum Spielen, Programmieren und Experimentieren. Viele Leser verwenden ihn als Hilfsmittel, um bestimmte Aufgaben leichter und schneller lösen zu können, zum Beispiel zur statistischen Auswertung beim Kegelclub (Alle Neune..., Seite 31) oder zum Protokollieren und Nachspielen von Schachpartien (Schachmeister, Seite 28). Sie können Ihre Freunde und Bekannten mit persönlichen und speziellen Kreuzworträtseln überraschen (Seite 40). Aber auch Musiker (Synthesizer, Seite 48) und Astronauten (Sternenhimmel, Seite 34) kommen auf Ihre Kosten, und angehende Millionäre sind mit der Finanzmathematik (Seite 20) auf dem besten Weg dorthin.

Grafik

Dem C 64 Grafik zu entlocken, ist eine anstrengende, aber auch ungeheuer interessante Aufgabe. Was auf diesem Gebiet von unseren Lesern gezaubert wurde, ist fast nicht zu glauben. HI-EDDI (Seite 58) schlägt wohl nicht nur alle Rekorde, sondern stellt auch die meisten vergleichbaren professionellen Mal- und Zeichenprogramme in den Schatten. Wenn Sie mehr ein Anhänger bewegter Grafik sind, sollten Sie sich Trickfilm (Seite 85) unbedingt anschauen. Natürlich fehlt auch nicht die Grafikerweiterung Turtle-Grafik (Seite 71).



Hilfsprogramme

Ein Meilenstein wurde gesetzt mit Hypra-Load (Seite 94). Das Laden von Programmen von der Floppy wird mit diesem sensationellen Programm stark beschleunigt. Etwas später kam Hypra-Save (Seite 124) hinzu und machte nun auch das Speichern von Programmen schneller. Um Schnelligkeit geht's auch beim Exsort (Seite 92), dem Sortierprogramm für alle Fälle, das mehr kann als nur einfach sortieren. Für Freunde von Sprachen gibt es gleich zwei Leckerbissen: Strubs (Seite 98), der Precompiler mit zusätzlichen Basic-Befehlen und ein Forth-Compiler (Seite 116).

Spiele

Anhänger von Abenteuerspielen, den Adventures, werden mit dem Grab des Pharaos (Seite 126) und in der Burg des Grauens (Seite 135) knifflige Situationen meistern müssen. Abwechslung bringt dann aber mit Sicherheit Apocalypse now (Seite 140), und wer Wert auf Geschicklichkeit legt, kann sich mit Springvogel vergnügen (Seite 148), ein Spiel, das nicht so schnell langweilig wird, weil Sie einfach eigene neue Bilder entwickeln können.

Listing-Service

Wer keine Zeit oder keine Lust hat, alle Programme selbst in mühevoller Kleinarbeit abzuschreiben, kann wieder auf den bewährten Leserservice zurückgreifen. Alle Programme befinden sich auf zwei Disketten mit der Bestellnummer L6 85 S6. Die Disketten sind nur zusammen erhältlich und kosten 34,90 Mark.

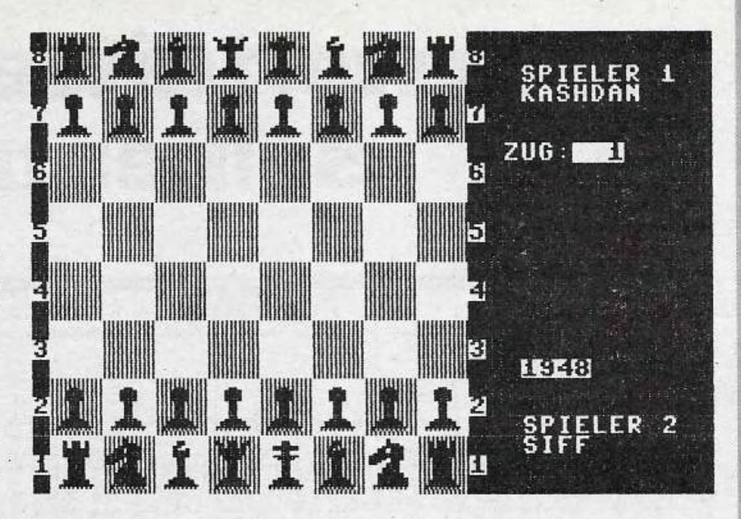
Bitte verwenden sie für Ihre Bestellung die beigefügte Postscheck-Zahlkarte zur Überweisung des Rechnungsbetrages. Sie erleichtern uns die Auftragsabwicklung und Sie sparen sich die 3 Mark Versandkosten.

Bestell-Nr.: L6 85 S6 / 34,90 Mark

Inhalt

Schachmeister

Darauf haben viele Schachspieler gewartet: Der C 64 nicht nur als Schachpartner, sondern auch als Protokollant eigener Schachpartien. Natürlich kann man sich auch große Meisterschaftspartien zu Gemüte führen. Besonders gut eignet sich der »Schachmeister« zum Einstudieren von Eröffnungen.



Jede beliebige Schachpartie kann am Bildschirm mit guter grafischer Darstellung nachvollzogen werden.

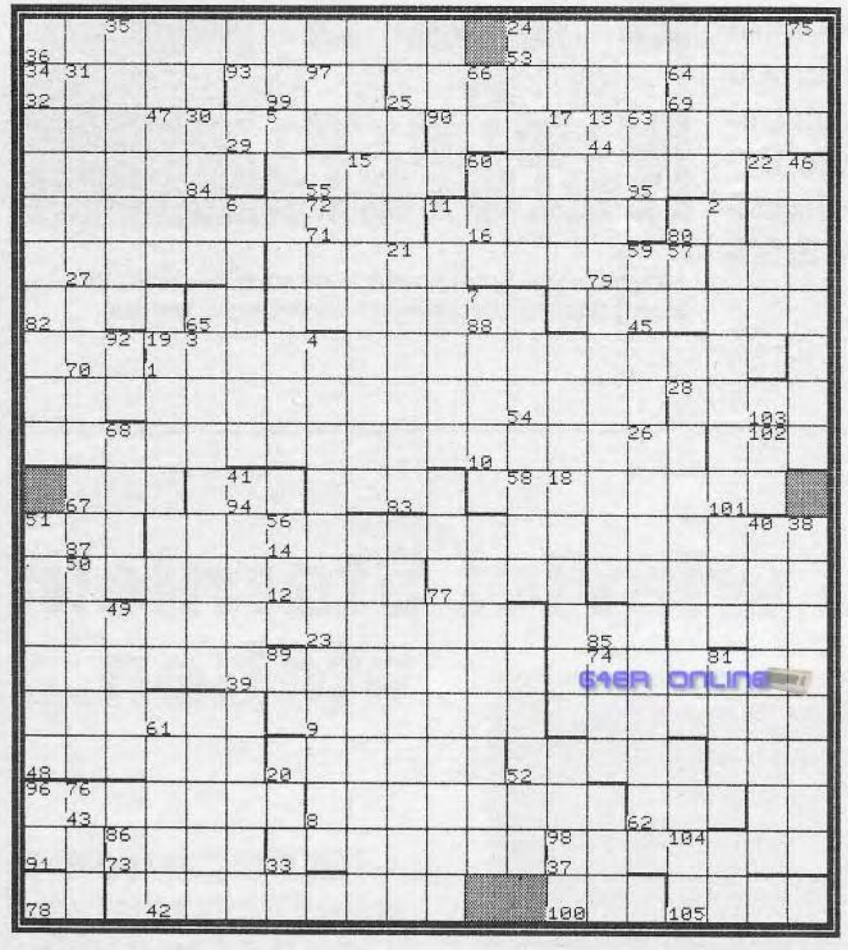
Weißt Du wieviel Sternlein stehen...



Mit diesem fantastischen Programm wird der C 64 zum Heim-Planetarium. Für jeden beliebigen Zeitpunkt und für jeden Ort der Erde kann in Sekundenschnelle eine naturgetreue elektronische Sternkarte erstellt werden. Sie enthält die hellsten und bekanntesten Fixsterne beziehungsweise Sternbilder des nördlichen Sternenhimmels.

Kreuzworträtsel selber machen

Dieses Programm, einst aus einem Wettbewerb entstanden, ist genau das richtige für Rätselsüchtige, die auch mal den umgekehrten Weg gehen wollen: ein Kreuzworträtsel von seiner Entstehung an mitverfolgen zu können. Sämtliche Schwierigkeiten die dabei auftreten können, werden vom C 64 übernommen. Sie können sich ganze Bibliotheken von Fachbegriffen aufbauen, so daß nachher Kreuzworträtsel entstehen, die auf einen ganz bestimmten Leserkreis zugeschnitten sind. Eine tolle Anwendung in einer Schülerzeitung oder für ein persönliches Rätsel für Freunde und Bekannte. Was ins Rätsel hineinkommt, ist ganz allein Ihre Sache.



Senkrecht:
(2) Im Dialog mit d. Computer arbeiten; (3) Speicherzelle mit direktem Zugriff; (4) Folge von Buchstaben/ Ziffern; (5) Programmdarstellung auf Papier; (6) Einheit des elektrischen Stroms; (7) Engl. f. 'Stapel'; (11) Tastenfolge b. engl. Tastaturen; (13) Menge der verwendbaren Symbole; (15) Sammelbegriff f. Computerbauteile; (17) Rueckkehr in den Anfangszustand; (18) Spannungs- oder Stromsignal; (19) Billiger Bandspeicher; (21) Jargon f. 'Diskettenstation'; (22) Ein Punkt auf dem Grafikbildschirm; (24) Hoehere strukturierte Progr.Sprache; (26) Vom Compiler erzeugter Kode; (28) Folge v. Anweisungen an d. Computer; (30) Basic-Befehl z. Laden des Speichers; (31) Variable z. Uebergabe an Unterprgr.; (34) Simulation eines fremden Computers; (35) Druckerteil f. Endlospapier; (38) Medium zum Festhalten von Daten; (40) Symbolische Einheit f.Wertzuweisung; (41) Parallele Drucker-Schnittstelle; (46) Logik mit diskreten Zustaenden; (47) Billiger Massenspeicher; (49) Logische Informationseinheit; (50) Symbolischer Name f. eine Adresse; (51) Einzelne logische Funktion; (56) Populaeres Betriebssytem f. Z80-uP; (57) Abk. f. 'Digital nach Analog'; (58) Verbindungsleitung im Prozessor; (59) Abk. f. 'Analog nach Digital'; (60) Hersteller von Mikrocomputern; (63) Steuerzeichen f. Kontrollcodes; (64) Mikroschalter auf der Platine; (66) ASCII-Zeichen f. Zeilenvorschub; (68) Verzweigungsanweisung in BASIC; (72) Strahlungseinheit; (74) Abk.f. 'hoeherwertiges Bit'; (75) Abk.f.'Exklusiv-Oder'-Verknuepfung; (76) Abk.f. 'Megahertz'; (81) Engl. f. Maedchen; (86) Neues Testament; (89) Evangelisch; (90) Grand Prix; (92) Volkswagen; (93) Deutscher Fussballbund; (96) Kurzbez. f. 'lichtjahr'; (97) Doktor; (98) Kurzbez. f. 'Technische Univers.'; (102) Firma; (104) Interessengemeinschaft

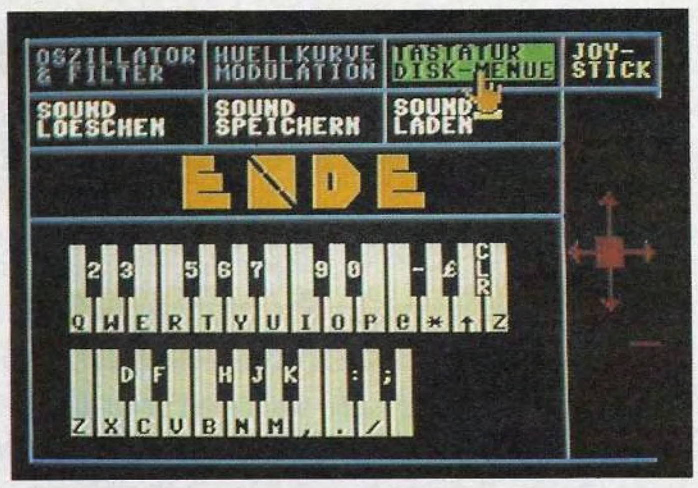
Waagerecht:
(1) Aufgabe dieses Wettbewerbs; (8) Wichtiges Eingabegeraet; (9) Programm zur Texteingabe/ Korrektur; (10) Schreibmarke auf dem Bildschirm; (12) Basic-Befehl zum Lesen d. Speichers; (14) Pruefsumme einer Bytefolge; (16) Leicht lernbare Progr.-Sprache; (20) Meldung ueber einen Geraetzustand; (23) Die haelfte eines Bytes oder 4 Bit; (25) Programm zum Starten von Computern; (27) Geraet z. Computerverb. per Telefon; (29) Einheit f. d. elektr. Kapazitaet; (32) Wiederverwendbares ROM; (33) Bildschirmausschnitt; (36) Datentransfer m. mehreren Leitungen; (37) Programm zur Steuerung v. Geraeten; (39) Sich selbst aufrufendes Unterprogr.; (42) Jargon fuer 'Computer'; (43) Punktfeld f. Zeichendarstellung; (44) Adresse die andere Adr. verweist; (45) Logische Informationseinheit; (48) Befehl zur Beendigung v. Unterprgr.; (52) Hoehere kommerzielle Prgr.Sprache; (53) Mech. Geraet z. Zeichnen v. Bildern; (54) Logische Speichereinheit; (55) Hochintegrierter Baustein; (62) Abk. f. 'Zeichen loeschen'; (65) Abk.f. 'Zeilendrucker'; (67) Technik zur Herstellung von IC's; (69) Abk. f.'Disketten-Betriebssystem'; (70) Abk. f. 'Television'; (71) Les- und schreibbarer Speicher; (73) Abk. f. 'Nanosekunde'; (77) Progr.Sprache f. Listenverarbeitung; (78) Abk. f. 'Hertz'; (79) Abk.f. Computer-gestuetzter Entwurf; (80) Griechischer Buchstabe; (82) Dritter Ton vom Grundton; (83) Abk. f. ein engl. Laengenmass; (84) Abk. f. 'Oberbuergermeister'; (85) In Ordnung; (87) Abk. f. 'Sankt'; (88) Seemeile; (91) Abk. f. 'Jahrhundert'; (94) Corps Consulaire; (95) Sozialdemokrat. Partei Deutschl.; (99) Norddeutscher Rundfunk; (100) Kurzbez. f. 'Universitaet'; (101) Kilo-Ampere; (103) Abk. f. 'Hektar'; (105) Abk. f. 'gegruendet'

Das Ergebnis eines Programmlaufs. Während das Programm läuft, werden die vom Computer eingesetzten Wörter direkt in das Rätsel am Bildschirm sichtbar eingetragen. Das geht so schnell, daß sie mit dem Auge kaum nachkommen können. Erst im Ausdruck werden alle Begriffe durch Zahlen ersetzt. Die Lösung diese Rätsels sehen Sie rechts.

Lösung des Kreuzworträtsels

```
bitparallel plottere
epromdndrurladerdoso
maabpfaradgfsrzeiger
yrkaobuchipcceespdpd
latnkasramqbasioxiii
amodemderfwmlecadnxg
terzlprmdlesmthdatei
itvkreuzwortraetselt
oewaerceaptabankprha
nrgsgekirpycursorafl
 mosiccceydkbiaboka
gstesecheckumtjgtvs
amottnpeeklispzeriap
tawtetmnibbleuokavre
troerrekursivlmtmgii
ekrpioveditorsskmiac
returnstatuscobolrbh
lmatrixtastaturdelle
jhnsecfenstertreiber
hztmaschine  unigegr
```

»Happysynth«, der Traum eines jeden Musikers



Eine kleine Hand (rechts oben im Bild) wird mit dem Joystick auf die gewünschte Funktion gebracht, und diese mit dem Feuerknopf angewählt.



Mit dem Untermenü »Oszillator & Filter« können alle nur erdenklichen Parameter per Joystick eingestellt werden.

Ein Supersynthesizer zum Abtippen. Die hervorragende Benutzerführung per »Pseudo-Maus« (siehe Fotos) und die ausgezeichnete grafische Darstellung der einzelnen Menüs heben den »Happysynth« von der breiten Masse der übrigen Programme ab.

Vier Pseudo-VICs mit 32 Sprites

Erweitern Sie Ihren Videochip. Nur duch Software lassen sich ungeahnte Bilder aus dem VIC zaubern. So sind 32 Sprites sowie HiRes-Grafik und Text gleichzeitig nur die leichtesten Übungen für Provic 64.

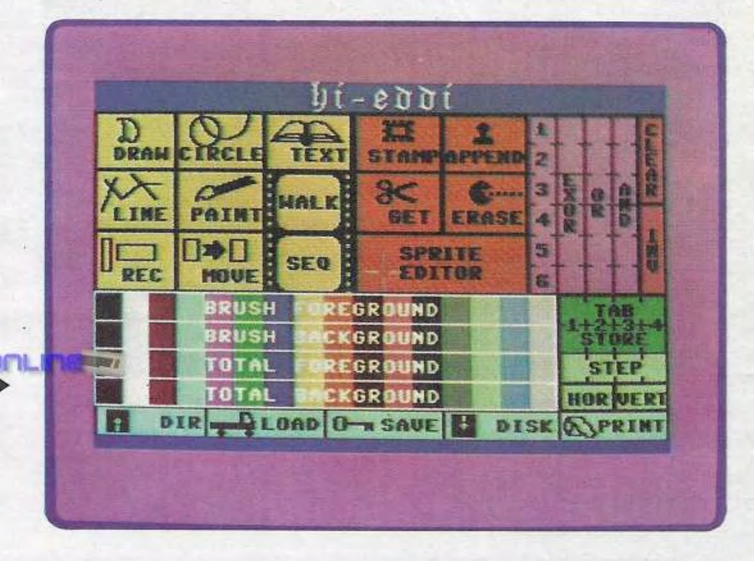


Provic 64 — Vier Bildschirmfenster mit bis zu 32 Sprites

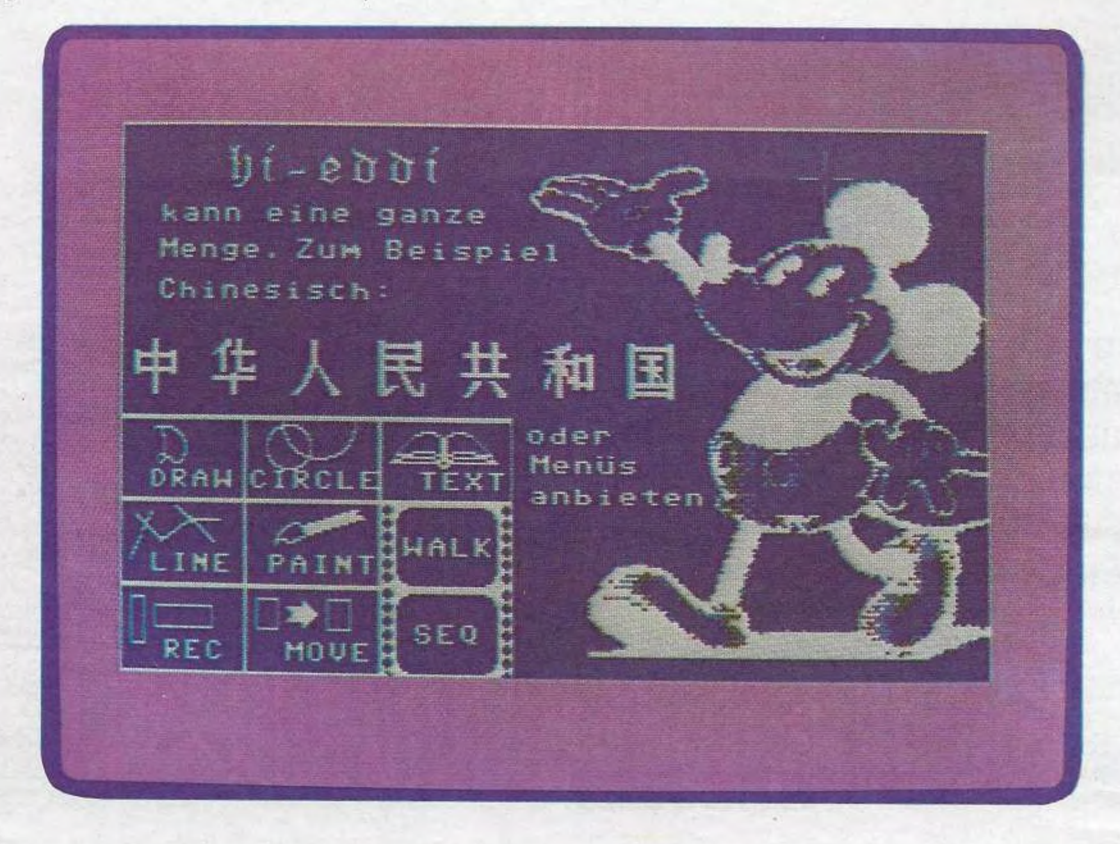
HI-EDDI: ein fantastisches Zeichen- und Malprogramm

»Ein professionelles Programm !« So urteilten unsere Leser nach der Veröffentlichung von HI-EDDI in der Ausgabe 1/85. Mit diesem Listing landeten wir einen echten Hammer! In diesem Sonderheft finden Sie nicht nur den normalen HI-EDDI, sondern auch Hardcopy-Routinen für alle gängigen Drucker sowie die komplette Anpassung für Datasetten-Besitzer gesammelt. Mit HI-Eddi kann jeder professionelle Bilder und Grafiken entwickeln.

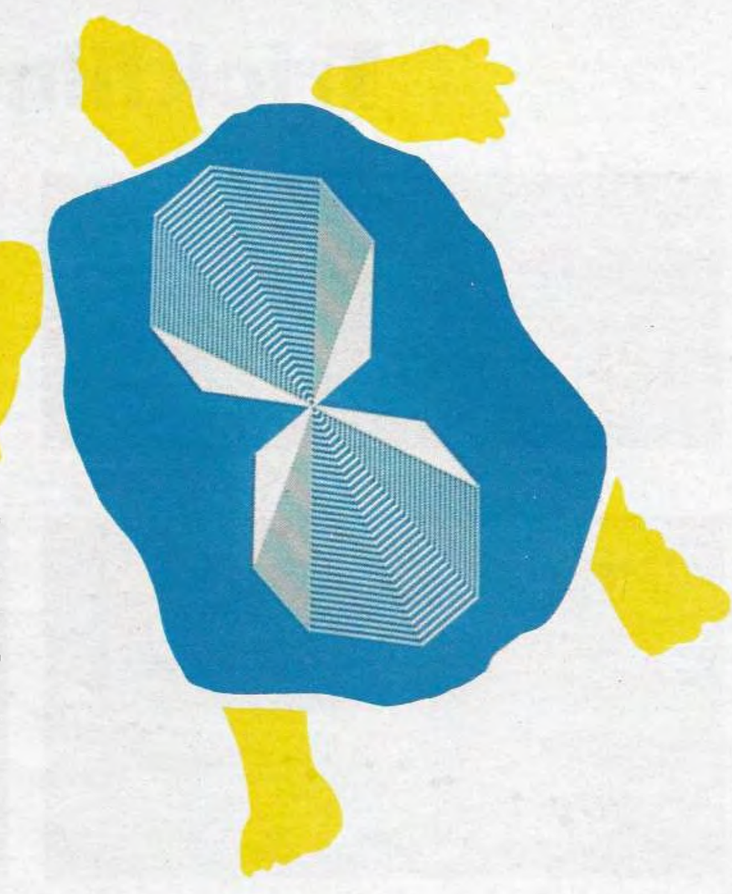
Solche oder auch ganz andere Menüs können Sie sich selbst erstellen. Auf der Diskette zu dieser Ausgabe finden Sie diese und noch andere Beispiele ▶



Es lassen sich nicht nur Bilder zeichnen, sondern auch technische Zeichnungen realisieren. Text kann ebenso eingefügt werden wie Sprites und selbstdefinierte Zeichensätze ▼

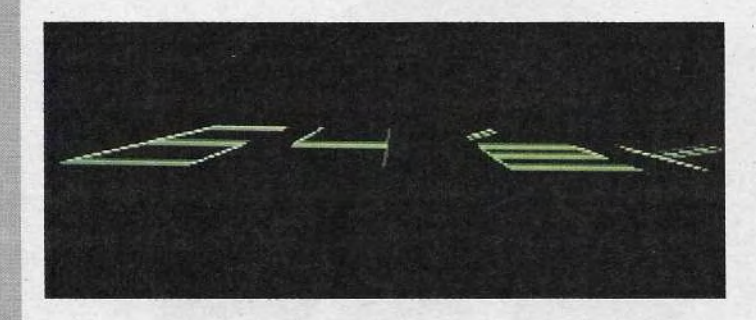


Die schnelle Schildkröte — Turtle Grafik

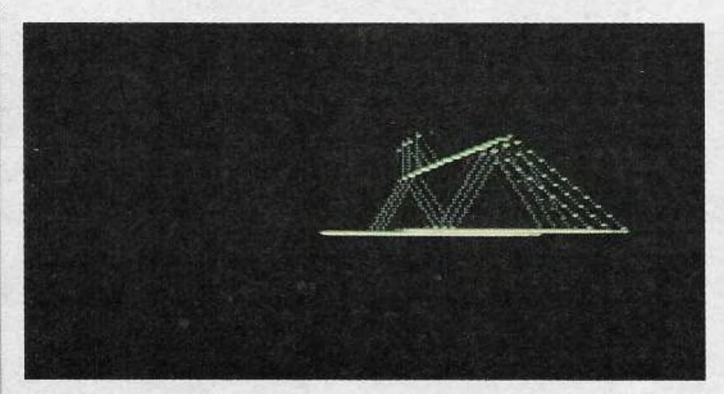


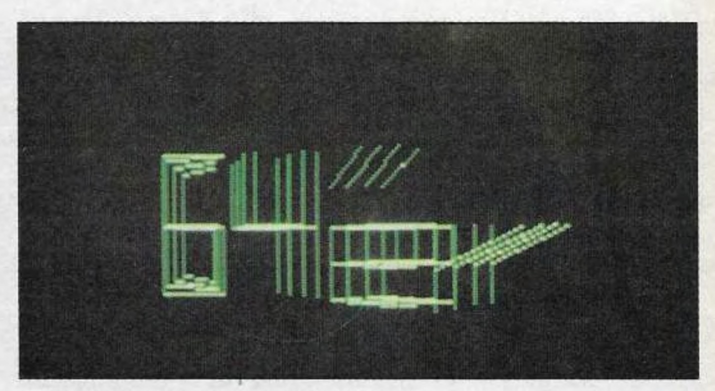
Eine Basic-Erweiterung, die Logo überflüssig macht, das ist die Turtle-Grafik. Mit einfachen Basic-Befehlen lassen sich sehr schnell wunderschöne Grafiken erstellen.

Trickfilm mit dem C 64



Die Fähigkeit, bewegte, dreidimensionale Grafiken zu erzeugen, war bisher nur speziellen Großcomputern vorbehalten. Mit unserem 3D-Movie-Maker wird auch Ihr Commodore 64 zum Trickfilmstudio.





Ein stufenloser Zoom-Effekt ist nur eine der vielen interessanten Möglichkeiten des 3D-Movie-Makers.



Hypra-Load und Hypra-Save

Ist Ihnen die 1541 zu langsam? Wollen Sie nicht mehrere hundert Mark für ein Hardware-Beschleunigungssystem ausgeben? Dann sind Hypra-Load und Hypra-Save genau das richtige für Sie, denn diese Programme machen das Laufwerk bis zu fünfmal schneller. Für dieses Sonderheft wurde das beliebte Hypra-Load sogar völlig neu geschrieben und verbessert!

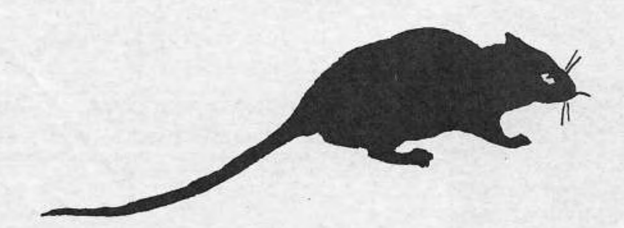

64'er

Das Grab des Pharao

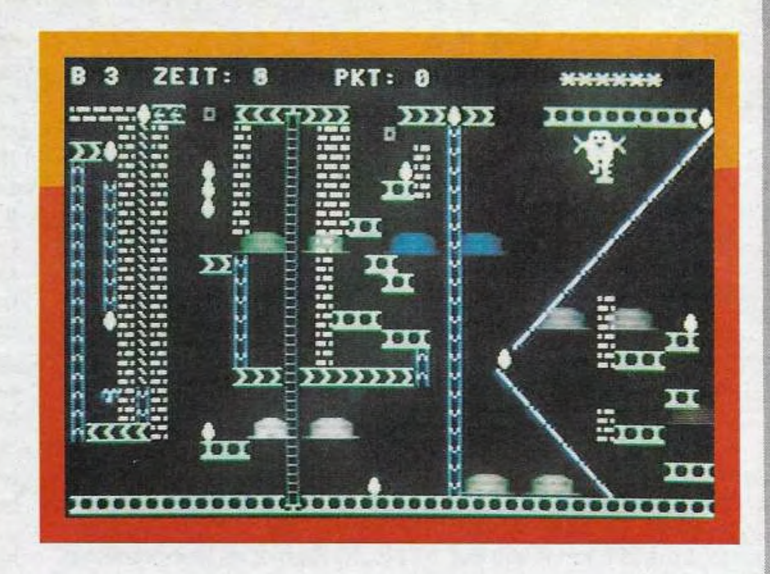
Hier bieten wir ihnen ein deutschsprachiges Grafik-Adventure der Spitzenklasse, bei dem es um Pyramiden, Pharaonen und viele Gefahren geht.

Castle of Doom — Burg des Grauens

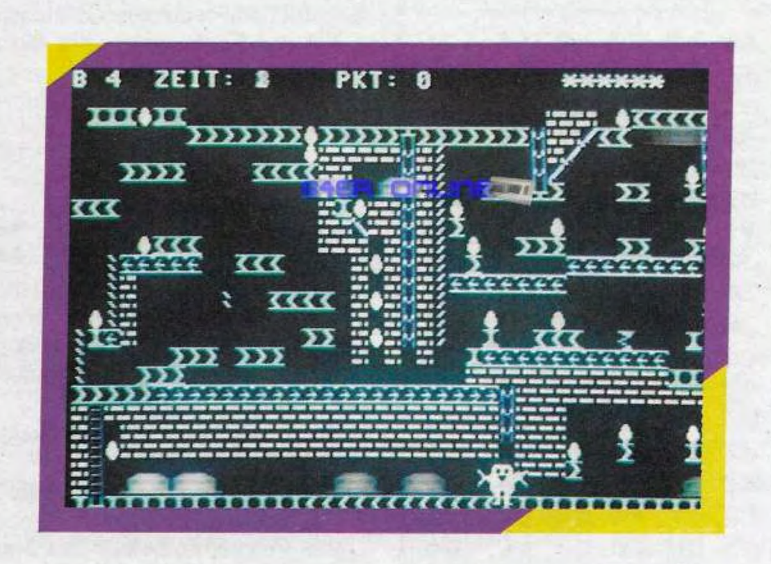
Nur mit Geduld und Fantasie wird es Ihnen gelingen sich die Lösung dieses Adventures zu erspielen. In der Burg des Grauens werden Sie sich die Zähne ausbeißen.

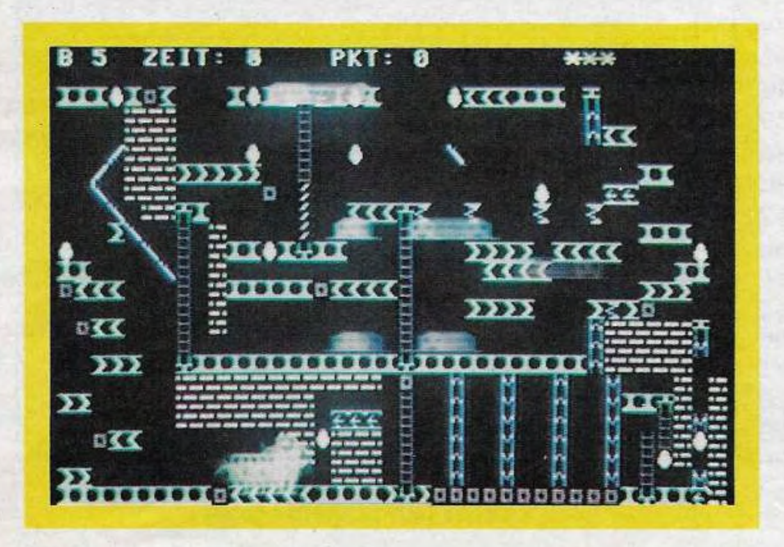


Dieses Spiel der Sonderklasse ist nur mit so legendären Programmen wie »Manic Miner«, »Jumpman« oder »Mister Robot and his Factory« vergleichbar. Sechs verschiedene Bilder garantieren andauerenden Spielspaß.



Spring Vogel, spring





Und sollten Sie alle Bilder durchgespielt haben, beginnt ein neues Kapitel: Machen Sie Ihre eigenen Bilder! Mit dem eingebauten Spiele-Editor ist das gar kein Problem. So wird der Spielspaß nur durch Ihre Fantasie begrenzt. Ein wirkliches Top-Programm.

Der Bitmap-Compander

Mit diesem kurzen Programm können Grafikbildschirme, die auf Diskette abgespeichert werden, auf die Hälfte komprimiert werden.

Jeder, der schon einmal eine hochauflösende Grafik auf Diskette abgespeichert hat, weiß, daß sie 33 Blocks benötigt. Das entspricht den 8 KByte des hochauflösenden Grafikbildschirms. Es besteht aber die Möglichkeit, diese Grafikbildschirme je nach Inhalt auf 10 bis 20 Blocks zu komprimieren. Das Verfahren beruht darauf, häufig wiederkehrende Sequenzen durch kürzere zu ersetzen. Dazu kann man sich die Bitmap als eine lange Kette von Nullen und Einsen vorstellen. Diese Kette wird jetzt in 4-Bit-Blöcke aufgeteilt. Solche Blöcke bezeichnet man als Halb-Byte oder Nibbles. Es gibt 16 verschiedene solcher Nibbles, wovon einige häufiger und andere seltener vorkommen. Jedem Nibble wird nun ein neuer Code zugeordnet, wobei das häufigste Nibble »0000«, den kürzesten Code »0« bekommt. Alle anderen 15 Codes müssen mit einer »1« beginnen. Einen Code »00« darf es nicht mehr geben, denn er könnte bei der Decodierung nicht mehr von dem Code »0« unterschieden werden. Es läßt sich nicht umgehen, viele Codes länger als 4 Bit zu machen. Die komplette Code-Tabelle ist in Tabelle 1 zu sehen. Doch die langen Codes kommen viel seltener vor als die kurzen, so daß bei einem »normalen« Bild viele Bits gespart werden können. Theoretisch wäre es möglich, daß eine komprimierte Bitmap länger wird als das Original. Aber selbst die Fotos aus der »Diashow« lassen sich komprimieren. Sir Winston Churchill magert zum Beispiel auf 24 Blocks ab. Bilder, wie sie üblicherweise in Gafik-Adventures vorkommen, lassen sich meist auf unter 15 Blocks zusammendrücken.

Tippen Sie das Programm (Listing 1) mit dem MSE ab und speichern Sie es, damit es jederzeit mit LOAD "BMC.EXE",8,1 geladen werden kann (auf der Leserservice-Diskette unter "BIT-MAPCOMPANDER" gespeichert). Der Aufruf des Companders funktioniert vom Basic aus mit einem SYS-Befehl. Die Parameter werden wie beim OPEN-Befehl angehängt.

Laden einer komprimierten Bitmap:

SYS 52798,Filenummer,Geräteadresse,Kanalnummer,"Name"

Speichern einer komprimierten Bitmap:

SYS 52736,Filenummer,Geräteadresse,Kanalnummer,"Name,P,W"

Mit der Filenummer wird die Anfangsadresse der Bitmap ausgewählt.

Filenummer	Bitmap-Adresse	
1	$2000 = 8192	(sichtbares Bild bei Hi-Eddi)
2	$4000 = 16384	
3	$6000 = 24576	
5	$A000 = 40960	(Bitmap bei Hires-3)
7	$E000 = 57344	(Bitmap bei Simons Basic)

Die Bitmaps 0, 4 und 6 sind theoretisch möglich, aber nicht sinnvoll, da der Videocontroller sie nicht adressieren kann. Die Gerätenummer ist 8 für das Floppy-Laufwerk und die Kanalnummer eine beliebige Zahl von 2 bis 14.

Beispiele für Aufrufe:

Speichern einer Simons Basic-Bitmap:

SYS 52736,7,8,2,"Name,P,W"

Laden dieser Bitmap in Hires-3:

SYS 52798,5,8,2,"Name"

Das Programm belegt den Bereich von $CE00 bis $CF19 (52736 bis 53017). Es belegt somit keinen Basic-Speicherplatz und verträgt sich auch mit Hires-3 und Simons Basic, solange der Befehl MEM nicht verwendet wird.

(Hans Haberl/ah)

```
 programm : bmc.exe          ce00 cf19
-----------------------------------
ce00 : 20 89 ce a6 b8 20 c9 ff   21
ce08 : a9 08 85 fc 78 a9 34 85   67
ce10 : 01 a0 00 b1 f8 a2 37 86   26
ce18 : 01 58 48 4a 4a 4a 4a 20   01
ce20 : 9d ce 68 29 0f 20 9d ce   6a
ce28 : e6 f8 d0 02 e6 f9 a5 f9   c7
ce30 : 29 1f c9 1f d0 d6 a5 f8   8b
ce38 : c9 40 d0 d0 f0 42 20 89   24
ce40 : ce a6 b8 20 c6 ff 20 cf   20
ce48 : ff 85 fa 20 cf ff 85 fb   d7
ce50 : a9 08 85 fc 20 bc ce 0a   35
ce58 : 0a 0a 0a 48 20 bc ce 68   e7
ce60 : 05 fd a0 00 78 a2 34 86   06
ce68 : 01 91 f8 a2 37 86 01 58   21
ce70 : e6 f8 d0 02 e6 f9 a5 90   3d
ce78 : f0 da a5 f8 c9 40 d0 d4   e9
ce80 : 20 cc ff a5 b8 20 c3 ff   56
ce88 : 60 20 fd ae 20 be e1 a5   18
ce90 : b8 0a 0a 0a 0a 0a 85 f9   0c
ce98 : a9 00 85 f8 60 a8 b9 09   06
cea0 : cf aa b9 e9 ce a4 fc 0a   8a
cea8 : 26 fa 88 d0 09 48 a5 fa   e7
ceb0 : 20 d2 ff a0 08 68 ca d0   de
ceb8 : ee 84 fc 60 a2 0f a5 fa   63
cec0 : 3d f9 ce dd e9 ce f0 03   48
cec8 : ca d0 f3 86 fd bd 09 cf   59
ced0 : aa a4 fc 06 fb 26 fa 88   ba
ced8 : d0 07 20 cf ff 85 fb a0   8b
cee0 : 08 ca d0 ef 84 fc a5 fd   42
cee8 : 60 00 a0 a8 b0 b8 c0 c8   eb
cef0 : f0 d0 d8 e0 f4 e8 f8 fc   0f
cef8 : 80 80 f8 f8 f8 f8 f8 f8   43
cf00 : fc f8 f8 f8 fc f8 fc fc   5b
cf08 : e0 01 05 05 05 05 05 05   e1
cf10 : 06 05 05 05 06 05 06 06   28
cf18 : 03 ff fc 68 ff ff ff 68   38
```

Listing 1. zum Programm »Bitmap-Compander«. Bitte mit dem MSE eingeben.

Nibbles	Codes	Hex	Länge
0000	0	$00	1
0001	10100	$A0	5
0010	10101	$A8	5
0011	10110	$B0	5
0100	10111	$B8	5
0101	11000	$C0	5
0110	11001	$C8	5
0111	111100	$F0	6
1000	11010	$D0	5
1001	11011	$D8	5
1010	11100	$E0	5
1011	111101	$F4	6
1100	11101	$E8	5
1101	111110	$F8	6
1110	111111	$FC	6
1111	100	$80	3

Tabelle 1. Tabelle aller Nibbels und möglichen Codes (mit Längenangabe)



Checksummer 64 — Neu

Der Checksummer 64 V3 überprüft jede Basic-Zeile direkt nach der Eingabe, erkennt Fehleingaben und auch Vertauschungen von Zahlen und Ziffern, und erspart deshalb eine aufwendige Fehlersuche.

Der Checksummer 64 V3 ist ein kleines Maschinenprogramm, das Sie sofort unterrichtet, ob Sie die jeweilige Programmzeile korrekt eingegeben haben.

So gehen Sie vor:

1. Programm abtippen und speichern.

2. Starten mit RUN

3. Nach kurzer Zeit sehen Sie am Bildschirm:
CHECKSUMMER 64, CHECKSUMMER AKTIVIERT, AUSSCHALTEN MIT POKE 1,55, ANSCHALTEN MIT POKE 1,53, READY.

4. Anschalten des Checksummer 64 V3 mit POKE 1,53.

5. Test: Geben Sie in einer freien Zeile ein: »1 REM« und drücken die RETURN-Taste. Am Bildschirm oben links sollten Sie die Prüfsumme <63> sehen.

6. Geben Sie ein Listing aus unserem Heft ein. Nach jeder Zeile wird die Zahl, die im Listing in Klammern < > steht, in den Bildschirm eingeblendet. Stimmen die Zahlen nicht überein, so liegt vermutlich ein Eingabefehler vor. **Die Zahl in den Klammern, und auch die Klammern selbst, dürfen beim Abtippen nicht mit eingegeben werden!**

7. Dieser neue Checksummer 64 V3 bemerkt, im Gegensatz zu den bisherigen, auch Vertauschungen von Zahlen und Buchstaben.

8. Unsere Basic-Listings enthalten keine Steuerzeichen mehr. Diese werden ersetzt durch Klartext und stehen zwischen geschweiften Klammern. Deshalb sind weder die Klammern noch was dazwischen steht, abzutippen, sondern die in Tabelle 1 aufgeführten Tasten zu drücken. Auf Ihrem Bildschirm erhalten Sie dann wieder die entsprechenden Grafikzeichen (siehe Bild 1 und 2).

9. Alle Grafikzeichen werden ebenfalls ersetzt durch unterstrichene oder überstrichene Großbuchstaben. **Unterstrichene Buchstaben bedeuten, daß Sie die SHIFT-Taste und den angegebenen Buchstaben drücken müssen, überstrichene jedoch die Commodore-Taste mit dem Buchstaben.** Auch hier erhalten Sie am Bildschirm das entsprechende Grafikzeichen und nicht etwa das im Listing erkennbare Zeichen (siehe Bild 1 und 2).

Checksummer VC 20 V3

Der Checksummer VC 20 V3 ist im Prinzip genauso aufgebaut wie der Checksummer 64. Da beim VC 20 jedoch nicht die Möglichkeit besteht, das ROM softwaremäßig zu modifizieren, mußte ein anderer Weg als beim Commodore 64 gewählt werden, um die Checksumme zu generieren.

In ihrer Funktionsweise unterscheiden sich der Checksummer VC 20 und der Checksummer 64 nicht. Es gelten folgende Sonderregelungen bei der Benutzung des Checksummer VC 20:

— Da der Basic-Bereich nicht belegt werden soll, ist das Programm im Kassettenpuffer abgelegt.

— Angeschaltet wird der Checksummer VC 20 mit »SYS 955«.

— Abschaltung des Checksummer VC 20 wird mit »SYS 58459« vollzogen.

Achtung: Nehmen Sie keine Kassetten-Operationen vor, wenn der Checksummer VC 20 eingeschaltet ist. Da das Betriebssystem den Kassettenpuffer mit Daten belegt, kann der Checksummer VC 20 überschrieben werden, was zur Folge hat, daß sich der Computer bei aktiviertem Checksummer VC 20 »aufhängt«. Wollen Sie deshalb ein Programm auf (von) Kassette abspeichern (laden), so müssen Sie erst den Checksummer VC 20 abschalten (SYS 58459).

Daraufhin kann der Kassettenpuffer mit Daten überschrieben werden, ohne daß der Computer »aussteigt«.

Als Sicherung wird bei der Initialisierung geprüft, ob das zuletzt angesprochene Peripherie-Gerät der Kassettenrecorder war. Ist das der Fall, so werden die Betriebssystemroutinen LOAD und SAVE für die Benutzung gesperrt. Der Rechner meldet bei Aufruf einer dieser beiden Routinen READY, ohne weitere Aktionen durchzuführen. Diese Sicherung kann man nach der Tipparbeit aufheben, wenn man den Checksummer VC 20 mit SYS 58459 abschaltet. Dadurch wird der Kassettenpuffer für andere Daten freigemacht. Weiterhin wird dann durch gleichzeitiges Drücken der Tasten »Run-Stop & Restore« erreicht, daß die Betriebssystemroutinen LOAD und SAVE wieder eingerichtet werden.

— Bei Benutzung einer Diskettenstation brauchen Sie nicht darauf zu achten, daß bei LOAD beziehungsweise SAVE der

CTRL steht für Control-Taste, so bedeutet [CTRL-A], daß Sie die Control-Taste und die Taste »A« drücken müssen. Im folgenden steht:

{DOWN}	Taste neben rechtem Shift, Cursor unten
{UP}	Shift-Taste & Taste neben rechtem Shift; Cursor hoch
{CLR}	Shift-Taste & 2. Taste ganz rechts oben
{INST}	Shift-Taste & Taste ganz rechts oben
{HOME}	2. Taste von ganz rechts oben
{DEL}	Taste ganz rechts oben
{RIGHT}	Taste ganz rechts unten
{LEFT}	Shift-Taste & Taste unten rechts
{SPACE}	Leertaste
{F1}	grauer Tastenblock rechts
{F3}	grauer Tastenblock rechts
{F5}	grauer Tastenblock rechts
{F7}	grauer Tastenblock rechts
{F2}	grauer Tastenblock rechts & Shift
{F4}	grauer Tastenblock rechts & Shift
{F6}	grauer Tastenblock rechts & Shift
{F8}	grauer Tastenblock rechts & Shift
{RETURN}	Shift-Taste & Return
{BLACK}	Control-Taste & 1
{WHITE}	Control-Taste & 2
{RED}	Control-Taste & 3
{CYAN}	Control-Taste & 4
{PURPLE}	Control-Taste & 5
{GREEN}	Control-Taste & 6
{BLUE}	Control-Taste & 7
{YELLOW}	Control-Taste & 8
{RVSON}	Control-Taste & 9
{RVOFF}	Control-Taste & 0
{ORANGE}	Commodore-Taste & 1
{BROWN}	Commodore-Taste & 2
{LIG.RED}	Commodore-Taste & 3
{GREY 1}	Commodore-Taste & 4
{GREY 2}	Commodore-Taste & 5
{LIG.GREEN}	Commodore-Taste & 6
{LIG.BLUE}	Commodore-Taste & 7
{GREY 3}	Commodore-Taste & 8

Wenn Sie sich erst einmal an die in Klartext geschriebenen Steuerzeichen gewöhnt haben, werden Sie den Vorteil dieser Schreibweise erkennen. Der zu dem jeweiligen Steuerzeichen gehörende Klartext ist so verfaßt, daß Sie leicht die Taste beziehungsweise die Tastenkombination finden, die Sie drücken müssen.

Tabelle 1. Die Steuerbefehle im Klartext

Checksummer VC 20 überschrieben wird, da der Kassettenpuffer für die Diskettenstation normalerweise nicht genutzt wird. Deshalb können Sie die beiden Routinen weiterhin normal nutzen, sofern der Rechner bei der Initialisierung des Checksummer VC 20 feststellt, daß das zuletzt angesprochene Peripherie-Gerät nicht der Kassettenrecorder war.
— Bedingt durch den anderen Aufbau des Checksummer VC 20 wird anders als beim Checksummer 64 nach der LOAD-Routine keine Checksumme ausgegeben.
— Wird eine Zeile gelöscht, also eine Zahl zwischen 0 und 65999 eingegeben, und danach Return gedrückt, so wird eine Checksumme ausgegeben, die aber keine Bedeutung hat.

Sie können die Programme auch weiterhin ohne den Checksummer eintippen. (F. Lonczewski/gk)

Hinweis: {13 SPACE} bedeutet 13mal die Leertaste drücken

```
1 REM ********************************
2 REM *                              *
3 REM *      CHECKSUMMER 64 V3       *
4 REM *                              *
5 REM *    WRITTEN MAERZ 1985 BY     *
6 REM *                              *
7 REM *      FRANK LONCZEWSKI        *
8 REM *                              *
9 REM ********************************
10 PRINT"{CLR,11SPACE,RVSON}CHECKSUMMER 64
   V3{RVOFF}"
11 PRINT"{2DOWN,9SPACE}EINEN MOMENT, BITTE
   ..."
12 FOR I=828 TO 864:READ A:POKE I,A:PS=PS+
   A+1:NEXT I
13 IF PS<>5802 THEN PRINT"PRUEFSUMMENFEHLE
   R IN ZEILEN 20-22":END
14 SYS 828:PS=0:FOR I=58464 TO 58583:READ
   A:POKE I,A:PS=PS+A+1:NEXT I
15 IF PS<>16267 THEN PRINT"PRUEFSUMMENFEHL
   ER IN ZEILEN 22-30":END
16 POKE 1,53:POKE 42289,96:POKE 42290,228
17 PRINT"{4DOWN,9SPACE}CHECKSUMMER AKTIVIE
   RT."
18 PRINT"{2DOWN}AUSSCHALTEN : POKE1,55"
19 PRINT"{DOWN}ANSCHALTEN{2SPACE}: POKE1,5
   3":NEW
20 DATA 169,0,133,254,162,1,189,93,3,133,2
   55,160,0,177,254
21 DATA 145,254,136,208,249,230,255,165,25
   5,221,95,3,208,238,202
22 DATA 16,230,96,160,224,192,0,160,2,169,
   0,170,133,254,177
23 DATA 95,240,40,201,32,208,3,200,208,245
   ,133,255,138,41,7
24 DATA 170,240,14,72,165,255,24,42,105,0,
   202,208,249,133,255
25 DATA 104,170,232,165,255,24,101,254,133
   ,254,76,111,228,192,4
26 DATA 48,219,198,214,165,214,72,162,3,16
   9,32,157,1,4,189
27 DATA 212,228,32,210,255,208,12,0,92,72,
   32,201,255,170,104
28 DATA 144,1,138,96,202,16,228,166,254,16
   9,0,32,205,189,169
29 DATA 62,32,210,255,104,133,214,32,108,2
   29,169,141,32,210,255
30 DATA 76,128,164,9,60,18,19
```

© 64'er

Dieser neue Checksummer 64 V3 erkennt auch Vertauschungen von Zahlen.

```
10 REM***************
11 REM*             *
12 REM* CHECKSUMMER *
13 REM*             *
14 REM* V3     VC20 *
15 REM*             *
16 REM*   WRITTEN   *
17 REM* MAERZ  1985 *
18 REM*     BY      *
19 REM*F. LONCZEWSKI*
20 REM***************
21 PRINT"{CLR,SPACE,RVSON}CHECKSUMMER V3 V
   C-20{RVOFF}"
22 PRINT"{2DOWN}EINEN MOMENT, BITTE..."
23 FOR I=827 TO 1019:READ A:POKE I,A
24 PS=PS+A+1:NEXT I
25 IF PS<>24464 THEN PRINT"{DOWN}PRUEFSUMM
   ENFEHLER !":END
26 SYS 981:PRINT"CHECKSUMMER AKTIVIERT."
27 PRINT"AN :SYS981"
28 PRINT"{DOWN}AUS:SYS58459, BEI CAS-{4SPA
   CE}SETTE ZUSAETZLICH{5SPACE}RUN/STOP &
   RESTORE"
29 PRINT"{DOWN}BEI AKTIVIERTEM CHECK-SUMME
   R KEIN";
30 PRINT" CASSETTEN-BETRIEB (LOAD, SAVE){2
   SPACE}ERLAUBT!":NEW
31 DATA 32,95,3,134,122,132,123,32,115,0,1
   70,240,243,162,255
32 DATA 134,58,144,10,162,0,134,255,32,121
   ,197,76,225,199,162
33 DATA 1,134,255,76,156,196,166,255,224,1
   ,240,3,76,96,197
34 DATA 160,2,169,0,170,133,254,177,95,240
   ,40,201,32,208,3
35 DATA 200,208,245,133,253,138,41,7,170,2
   40,14,72,165,253,24
36 DATA 42,105,0,202,208,249,133,253,104,1
   70,232,165,253,24,101
37 DATA 254,133,254,76,119,3,192,4,48,219,
   198,214,165,214,72
38 DATA 162,3,169,32,157,1,4,189,209,3,32,
   210,255,202,16
39 DATA 242,166,254,169,0,32,205,221,169,6
   2,32,210,255,104,133
40 DATA 214,32,135,229,169,141,32,210,255,
   162,0,134,255,240,148
41 DATA 9,60,18,19,169,59,141,2,3,169,3,14
   1,3,3,165
42 DATA 186,201,1,208,16,169,116,141,48,3,
   141,50,3,169,196
43 DATA 141,49,3,141,51,3,173,136,2,141,17
   0,[illegible]5
```

© 64'er

Der neue Checksummer VC 20 V3 erkennt auch Vertauschungen von Zahlen.

```
5 PRINT CHR$(14)                                  <242>
10 PRINT"{CLR}"                                   <254>
20 PRINT"Q****************************W"          <130>
30 PRINT"{4DOWN,2SPACE}TEST{SPACE,BLUE,6SP
   ACE}"                                          <022>
40 PRINT"BBBBBBBBBBBBBBBBBBBBBBBBBBBBB"           <108>
```

© 64'er

Bild 1. So könnte ein Teil eines Listings abgedruckt sein. In Zeile 10 müssen Sie nach den Anführungsstrichen die CLEAR/HOME-Taste drücken und nicht die Klammern mit dem Wort CLR. In Zeile 20 drücken Sie nach den Anführungsstrichen die Commodore-Taste und den Buchstaben Q, gefolgt von mehreren SHIFT und Stern-Taste, und zum Schluß die Commodore-Taste und den Buchstaben W. In Zeile 30 ist es viermal die Cursor-nach-unten-Taste, gefolgt von zweimal die Leertaste, dann SHIFT und T und normal EST, zum Schluß noch einmal die Leertaste, die Farbtaste Blau (Control und 7) und sechsmal die Leertaste. Zeile 40 besteht lediglich aus mehreren Grafikzeichen, die mit der Commodore-Taste und B erzeugt werden.

```
5 PRINTCHR$(14)
10 PRINT"□"
20 PRINT"├──────────────────────────┤"
30 PRINT"▒▒▒▒  |EST ▒      "
40 PRINT"▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚"
```

Bild 2. Auf dem Bildschirm oder Ihrem Drucker sieht das Listing (Bild 1) so aus.

MSE – Abtippen sicher und leicht gemacht

Ähnlich wie der »Checksummer« ist auch der MSE ein Hilfsmittel bei der Eingabe von Listings, diesmal jedoch bei reinen Maschinensprache-Programmen.

Im Gegensatz zum »Checksummer« aber ist die Eingabe nicht ohne den MSE möglich. Der MSE verringert die Tipparbeit um ein Drittel und schließt Fehleingaben vollkommen aus. Außerdem können Sie die Werte blind eingeben, ohne andauernd auf den Bildschirm schauen zu müssen. Dies wird durch akustische Meldungen realisiert.

MSE ist ein Maschinenspracheditor, mit dem ein Vertippen ausgeschlossen ist. Eine abgetippte Zeile wird nur angenommen, wenn sie richtig ist. Eine Checksumme am Ende jeder Zeile prüft, ob die richtigen Werte in der richtigen Zeile an der richtigen Stelle stehen. Wenn nicht, ertönt ein Warnsignal, und man beseitigt den Fehler.

War die Zeile korrekt, erklingt ein Gong, und die nächste Zeilennummer wird ausgegeben. Damit ist also auch »blindes« Eintippen möglich; Sie können sich voll auf den Text konzentrieren.

So arbeitet man mit MSE

Laden und starten Sie MSE. Zuerst wird der Programmname und die Start- und Endadresse erfragt. **Diese Angaben entnehmen Sie dem Kopf des jeweiligen abgedruckten Listings.** MSE meldet sich dann mit der Zeilennummer der ersten Zeile. Wenn Sie die Zeile richtig eingegeben haben, erscheint die nächste Zeilennummer und so weiter bis zum Ende. Zum Schluß wird das fertige Programm mit »CTRL-S« auf Diskette oder Kassette abgespeichert. Dazu sind keine weiteren Angaben mehr erforderlich. Das Programm kann dann ganz normal wieder geladen und gestartet werden. Wenn Sie nicht alles auf einmal tippen wollen, können Sie jederzeit unterbrechen und den eingetippten Teil mit »CTRL-S« abspeichern. Wollen Sie weiterarbeiten, laden und starten Sie MSE wieder.

Geben Sie auf die Frage nach der Startadresse aber jetzt »L« ein, um Ihr Teilprogramm zu laden. Jetzt können Sie mit »CTRL-N« die Adresse eingeben, an der Sie weitertippen müssen. Wenn Sie sich nicht gemerkt haben, wie weit Sie gekommen sind, geben Sie nach dem Laden »CTRL-M« ein.

Auf die Frage nach der Startadresse antworten Sie mit der Anfangsadresse, die links in der Kopfzeile auf dem Bildschirm steht. Nun wird Ihr Programm aufgelistet. Mit »SPACE« wird das Listen fortgesetzt, mit »STOP« abgebrochen. Das Ende Ihres Programmteils erkennen Sie sehr einfach daran, daß nur noch der Wert »AA« in der Zeile steht. Die Adresse dieser Zeile müssen Sie anschließend mit »CTRL-N« eingeben. Das Programm ist nur mit »STOP/RESTORE« zu verlassen. Speichern Sie aber vorher unbedingt immer Ihren Text ab.

Hinweise zum Abtippen

Vor dem Abtippen oder späteren Wiederladen des MSE-**Laders** müssen Sie unbedingt folgende Zeile eingeben:

POKE 43,1: POKE 44,32: POKE 8192,0: NEW

Starten Sie das Programm mit RUN. Fehlerhafte Zeilen werden angezeigt und müssen korrigiert werden, bis der Lader zum »READY« durchläuft. Jetzt müssen Sie das fertige MSE-Programm abspeichern. Dazu brauchen Sie nur »RETURN« zu drücken, weil die erforderlichen Angaben schon auf dem Bildschirm stehen. (Kassettenbesitzer müssen in Zeile 343 die letzte Zahl in »1« abändern.) Ab jetzt können Sie »MSE V1.0« direkt, also ohne den DATA-Lader, benutzen. MSE V1.0 wird ganz normal mit »,8« geladen (keine POKEs notwendig).

(N. Mann / D. Weineck / gk)

MSE-Befehle:

DEL	löscht die letzte Eingabe.
CTRL-S	speichert das eingetippte Programm ab.
CTRL-L	lädt ein Programm. Start- und Endadresse werden automatisch ermittelt.
CTRL-M	listet den Speicherinhalt. Abbruch mit STOP-Taste, weiter mit Leertaste.
CTRL-N	erlaubt die Eingabe einer neuen Adresse zum Weitertippen.
CTRL-P	gibt ein MSE-Listing auf dem Drucker aus.

```
100 REM *****************************        <091>
110 REM *                           *        <159>
120 REM *        M S E   LADER      *        <206>
130 REM *                           *        <179>
220 REM *****************************        <211>
230 REM                                      <036>
240 DIM H(75): FOR I=0 TO 9                  <113>
250 H(48+I)=I: H(65+I)=I+10:NEXT             <041>
260 FOR I=2048 TO 3755 : READ A$             <198>
270 H=ASC(LEFT$(A$,1)):L=ASC(RIGHT$(A$,1))  <199>
280 D=H(H)*16+H(L):S=S+D:POKE I,D            <219>
290 A=A+1:IF A<20 THEN NEXT:A=-1             <141>
300 PRINT " ZEILE:";1000+Z;                  <011>
310 READ V :Z=Z+1:IF V=S THEN 330            <218>
320 PRINT"PRUEFSUMMENFEHLER !":STOP          <138>
330 IF A<0 THEN 341                          <221>
340 S=0:A=0:PRINT:NEXT                       <046>
341 PRINT"{CLR}PO43,1:PO44,8:PO45,172:PO46
    ,14                                      <010>
342 POKE 631,19:POKE 632,13:POKE 633,13:PO
    KE 198,3                                 <249>
343 PRINT"{3DOWN}SAVE"CHR$(34)"MSE V1.0"CH
    R$(34)",8                                <171>
344 END                                      <092>
350 REM :                                    <111>
360 REM        *****************             <149>
370 REM        *      DATA     *             <078>
380 REM        *****************             <169>
390 REM :                                    <151>
1000 DATA 00,0B,08,0A,00,9E,32,30,36,31,00
     ,00,00,A2,08,A9,36,85,A4,A9, 1247       <119>
1001 DATA 08,85,A5,A9,00,85,A6,A9,B0,85,A7
     ,A0,00,B1,A4,91,A6,C8,D0,F9, 2888       <054>
1002 DATA E6,A5,E6,A7,CA,D0,F2,A9,36,85,01
     ,4C,00,B0,20,D1,B1,A9,06,8D, 2787       <144>
1003 DATA 21,D0,A9,03,8D,20,D0,8D,86,02,A0
     ,B3,A9,74,20,FF,B1,A0,B3,A9, 2667       <237>
1004 DATA B9,20,FF,B1,A0,00,20,CF,FF,99,01
     ,02,C8,C9,0D,D0,F5,88,F0,D2, 2912       <217>
1005 DATA C0,0F,90,02,A0,0E,8C,00,02,20,EA
     ,B1,A0,B3,A9,CF,20,FF,B1,20, 2323       <013>
1006 DATA 8E,B4,85,FC,85,62,20,8E,B4,85,FB
     ,85,61,20,A7,B4,D0,20,A0,B3, 2864       <199>
1007 DATA A9,E5,20,FF,B1,20,8E,B4,85,60,20
     ,8E,B4,85,5F,20,A7,B4,D0,0A, 2624       <091>
1008 DATA A5,61,C5,5F,A5,62,E5,60,90,06,20
     ,43,B3,4C,3A,B0,A9,AA,A0,00, 2379       <167>
1009 DATA 91,FB,E6,FB,D0,02,E6,FC,20,3F,B2
```

Der MSE zum bequemen Abtippen von Assemblerprogrammen. Beachten Sie den Checksummer auf Seite 16.

```
      ,90,EF,4C,FB,B4,A2,02,86,58, 3118     <152>
1010 DATA A9,A6,A0,9D,20,F2,B1,20,E4,FF,F0
      ,FB,C9,30,90,0C,C9,47,B0,08, 2970     <231>
1011 DATA C9,3A,90,0B,C9,41,B0,07,C9,14,D0
      ,0F,4C,0B,B1,20,D2,FF,A6,58, 2322     <121>
1012 DATA 95,F7,C6,58,D0,D2,60,AE,8D,02,F0
      ,26,C9,0C,D0,03,4C,0B,B6,C9, 2685     <057>
1013 DATA 13,D0,03,4C,8B,B5,C9,0D,D0,03,4C
      ,BA,B4,C9,10,D0,03,4C,68,B5, 2282     <225>
1014 DATA C9,0E,D0,06,20,5F,B4,4C,64,B1,4C
      ,92,B0,A5,F9,20,02,B1,0A,0A, 2132     <208>
1015 DATA 0A,0A,85,F9,A5,F8,20,02,B1,05,F9
      ,60,C9,3A,90,02,69,08,29,0F, 1950     <092>
1016 DATA 60,A6,59,E0,08,90,1F,A6,58,E0,02
      ,B0,06,20,D2,FF,4C,8E,B0,C6, 2509     <188>
1017 DATA 59,A0,14,A9,92,20,F2,B1,CA,D0,FA
      ,84,57,68,68,4C,8B,B1,A6,D3, 2891     <197>
1018 DATA E0,08,B0,03,4C,92,B0,20,D2,FF,A6
      ,58,E0,02,90,09,C6,59,20,D2, 2468     <049>
1019 DATA FF,C6,58,D0,F9,4C,8E,B0,48,4A,4A
      ,4A,4A,20,59,B1,68,29,0F,C9, 2419     <035>
1020 DATA 0A,90,02,69,06,69,30,4C,D2,FF,A2
      ,FC,9A,20,D1,B1,20,48,B2,20, 2261     <073>
1021 DATA EA,B1,20,9F,B2,A5,FC,20,4E,B1,A5
      ,FB,20,4E,B1,20,ED,B1,A9,3A, 2860     <148>
1022 DATA A0,20,20,F2,B1,A9,00,85,59,20,8E
      ,B0,20,ED,B1,A4,59,20,EF,B0, 2530     <233>
1023 DATA 91,FB,C8,84,59,C0,08,90,EC,20,10
      ,B2,A9,12,20,D2,FF,20,8E,B0, 2657     <105>
1024 DATA 20,EF,B0,C5,FF,F0,0D,20,43,B3,A9
      ,14,A0,14,20,F2,B1,4C,A2,B1, 2665     <034>
1025 DATA A9,92,20,D2,FF,20,33,B2,20,E0,B2
      ,20,3F,B2,90,9F,4C,8B,B5,A9, 2648     <123>
1026 DATA 93,20,D2,FF,A2,00,A9,03,9D,00,D8
      ,9D,00,D9,9D,00,DA,9D,00,DB, 2476     <237>
1027 DATA E8,D0,EF,60,A9,0D,2C,A9,20,4C,D2
      ,FF,20,D2,FF,98,4C,D2,FF,20, 2965     <160>
1028 DATA E4,FF,F0,FB,60,84,5D,85,5C,A0,00
      ,B1,5C,F0,06,20,D2,FF,C8,D0, 3100     <077>
1029 DATA F6,60,A5,FB,85,5A,A0,00,84,5B,B1
      ,FB,18,65,5A,85,5A,90,02,E6, 2606     <156>
1030 DATA 5B,06,5A,26,5B,C8,C0,08,90,EC,A5
      ,5A,65,5B,85,FF,60,18,A5,FB, 2467     <219>
1031 DATA 69,08,85,FB,90,02,E6,FC,60,A5,FB
      ,C5,5F,A5,FC,E5,60,60,A0,B3, 3106     <183>
1032 DATA A9,FB,20,FF,B1,A0,01,B9,00,02,20
      ,D2,FF,CC,00,02,C8,90,F4,A9, 2692     <098>
1033 DATA 10,ED,00,02,AA,20,ED,B1,CA,D0,FA
      ,A5,62,20,4E,B1,A5,61,20,4E, 2453     <236>
1034 DATA B1,20,ED,B1,A5,60,20,4E,B1,A5,5F
      ,20,4E,B1,A9,9F,20,D2,FF,20, 2575     <038>
1035 DATA EA,B1,24,5E,10,01,60,A9,12,20,D2
      ,FF,A2,28,20,ED,B1,CA,D0,FA, 2646     <161>
1036 DATA A9,92,4C,D2,FF,A5,D6,C9,16,B0,01
      ,60,A9,A0,85,A4,A9,78,85,A6, 2945     <204>
1037 DATA A9,04,85,A5,85,A7,A2,13,A0,27,B1
      ,A4,91,A6,88,10,F9,CA,F0,19, 2671     <208>
1038 DATA 18,A5,A4,69,28,85,A4,90,02,E6,A5
      ,18,A5,A6,69,28,85,A6,90,E0, 2503     <251>
1039 DATA E6,A7,4C,B6,B2,A9,91,4C,D2,FF,A9
      ,0F,8D,18,D4,A9,00,8D,05,D4, 2776     <000>
1040 DATA A9,F7,8D,06,D4,A9,11,8D,04,D4,A9
      ,32,8D,01,D4,A9,00,8D,00,D4, 2413     <126>
1041 DATA A0,80,20,09,B3,A9,10,8D,04,D4,60
      ,A2,FF,CA,D0,FD,88,D0,F8,60, 2914     <240>
1042 DATA A9,0F,8D,18,D4,A9,2D,8D,05,D4,A9
      ,A5,8D,06,D4,A9,21,8D,04,D4, 2385     <119>
1043 DATA A9,07,8D,01,D4,A9,05,8D,00,D4,A0
      ,FF,20,09,B3,A9,20,8D,04,D4, 2250     <078>
1044 DATA A9,00,8D,01,D4,8D,00,D4,60,38,20
      ,F0,FF,8A,48,98,48,18,A0,06, 2179     <175>
1045 DATA A2,18,20,F0,FF,A0,B4,A9,0A,20,FF
      ,B1,20,12,B3,20,E4,FF,F0,FB, 2931     <093>
1046 DATA A2,1D,A9,14,20,D2,FF,CA,D0,FA,68
      ,A8,68,AA,18,4C,F0,FF,0D,0D, 2704     <088>
1047 DATA 0D,20,20,20,20,20,20,20,4D,41,53
      ,43,48,49,4E,45,4E,53,50,52, 1144     <216>
1048 DATA 41,43,48,45,20,2D,20,45,44,49,54
      ,4F,52,20,0D,0D,20,20,20,20, 1023     <038>
1049 DATA 20,20,20,20,56,4F,4E,20,4E,2E,4D
      ,41,4E,4E,20,26,20,44,2E,57, 1128     <206>
1050 DATA 45,49,4E,45,43,4B,00,0D,0D,0D,20
      ,20,20,50,52,4F,47,52,41,4D, 1102     <117>
1051 DATA 4D,4E,41,4D,45,20,3A,20,00,0D,0D
      ,20,20,20,53,54,41,52,54,41, 1073     <095>
1052 DATA 44,52,45,53,53,45,20,3A,20,24,00
      ,0D,0D,20,20,20,45,4E,44,41, 1014     <129>
1053 DATA 44,52,45,53,53,45,20,20,20,3A,20
      ,24,00,92,05,20,50,52,4F,47, 1171     <217>
1054 DATA 52,41,4D,4D,20,3A,20,00,12,20,20
      ,2A,2A,2A,20,46,41,4C,53,43, 1024     <027>
1055 DATA 48,45,20,45,49,4E,47,41,42,45,20
      ,2A,2A,2A,20,20,92,00,0D,0D, 1058     <098>
1056 DATA 2A,2A,2A,20,45,4E,44,45,20,2A,2A
      ,2A,00,13,05,20,20,12,44,92, 920      <148>
1057 DATA 49,53,4B,20,4F,44,45,52,20,12,54
      ,92,41,50,45,0D,00,13,20,20, 1151     <035>
1058 DATA 49,2F,4F,20,2D,20,46,45,48,4C,45
      ,52,00,20,D1,B1,20,48,B2,A0, 1606     <012>
1059 DATA B3,A9,CF,20,FF,B1,20,8E,B4,85,FC
      ,20,8E,B4,85,FB,C5,61,A5,FC, 3207     <251>
1060 DATA E5,62,90,23,A5,FB,C5,5F,A5,FC,E5
      ,60,B0,19,20,A7,B4,D0,14,60, 2860     <112>
1061 DATA 20,A7,B4,F0,0C,85,F9,20,A7,B4,F0
      ,05,85,F8,4C,EF,B0,68,68,20, 2749     <088>
1062 DATA 43,B3,4C,5F,B4,20,CF,FF,C9,4C,D0
      ,09,20,D1,B1,20,48,B2,4C,0B, 2372     <046>
1063 DATA B6,C9,0D,60,A9,00,85,5E,20,5F,B4
      ,20,EA,B1,20,0D,B5,24,5E,30, 2042     <120>
1064 DATA 05,20,E4,FF,F0,FB,20,E1,FF,F0,26
      ,20,9F,B2,24,5E,10,09,20,4E, 2435     <198>
1065 DATA B5,20,0D,B5,20,60,B5,20,33,B2,20
      ,3F,B2,90,D7,A0,B4,A9,28,20, 2190     <207>
1066 DATA FF,B1,20,E4,FF,C9,0D,D0,F9,A9,00
      ,85,5E,A5,61,85,FB,A5,62,85, 3056     <240>
1067 DATA FC,20,E0,B2,4C,64,B1,A5,FC,20,4E
      ,B1,A5,FB,85,FF,20,4E,B1,A9, 3003     <221>
1068 DATA 20,A0,3A,20,F2,B1,A0,00,20,ED,B1
      ,B1,FB,20,4E,B1,C8,C0,08,90, 2566     <070>
1069 DATA F3,20,ED,B1,24,5E,30,03,A9,12,2C
      ,A5,20,20,D2,FF,20,10,B2,A5, 2190     <059>
1070 DATA FF,20,4E,B1,A9,92,20,D2,FF,4C,EA
      ,B1,A9,FF,85,B8,85,B9,A9,04, 3073     <029>
1071 DATA 85,BA,20,C0,FF,A2,FF,4C,C9,FF,20
      ,CC,FF,A9,FF,4C,C3,FF,20,5F, 3315     <189>
1072 DATA B4,A9,80,85,5E,20,4E,B5,20,48,B2
      ,A2,24,A9,2D,20,D2,FF,CA,D0, 2596     <111>
1073 DATA FA,20,EA,B1,20,EA,B1,20,60,B5,4C
      ,C1,B4,20,B8,B5,A6,5F,A4,60, 2812     <015>
1074 DATA A9,61,20,D8,FF,B0,0A,20,B7,FF,29
      ,BF,D0,03,4C,FB,B4,A9,01,20, 2577     <201>
1075 DATA C3,FF,20,68,B6,A0,B4,A9,4F,20,FF
      ,B1,20,F9,B1,4C,FB,B4,20,68, 2921     <237>
1076 DATA B6,A9,37,A0,B4,20,FF,B1,20,F9,B1
      ,A2,08,C9,44,F0,06,A2,01,C9, 2717     <213>
1077 DATA 54,D0,F1,A9,01,A8,20,BA,FF,A0,00
      ,E0,01,F0,1A,A9,40,8D,20,02, 2403     <101>
1078 DATA A9,3A,8D,21,02,B9,01,02,99,22,02
      ,C8,CC,00,02,90,F4,C8,C8,D0, 2182     <127>
1079 DATA 0C,B9,01,02,99,20,02,C8,CC,00,02
      ,D0,F4,98,A2,20,A0,02,4C,BD, 2018     <025>
1080 DATA FF,20,B8,B5,A5,BA,C9,08,90,33,A6
      ,B9,86,57,A9,01,20,C3,FF,A9, 2800     <022>
1081 DATA 60,85,B9,20,C0,FF,B0,28,A5,BA,20
      ,B4,FF,A5,B9,20,96,FF,20,A5, 2911     <053>
1082 DATA FF,85,61,A5,90,4A,4A,B0,13,20,A5
      ,FF,85,62,20,AB,FF,A5,57,85, 2663     <214>
1083 DATA B9,A9,00,20,D5,FF,90,03,4C,A3,B5
      ,86,5F,84,60,A5,BA,C9,01,D0, 2639     <131>
1084 DATA 0A,AD,3D,03,85,61,AD,3E,03,85,62
      ,4C,FB,B4,A9,13,20,D2,FF,A2, 2300     <120>
1085 DATA 1C,20,ED,B1,CA,D0,FA,60, 1230     <214>
```


MSE (Schluß). Dieses Listing können Sie (müssen aber nicht) mit dem neuen Checksummer 64 V3 in diesem Heft eingeben.

Menü-gesteuerte Finanz-mathematik

Zwei herausragende Eigenschaften machen dieses Programm für alle C 64-Besitzer interessant: Die sehr gute Menütechnik und der modulare Bildschirmmaskenaufbau.

Das Programm (Listing 1) wird nach dem Abtippen und Speichern aller Teilprogramme mit dem Befehl: LOAD"Menü" ,8 geladen.

Wählen Sie nach dem Start mit den Funktionstasten F3 und F5 die Nummer des Programmes, mit welchem Sie als nächstes arbeiten wollen (Zinsrechnung=Listing 2; Zinseszinsrechnung=Listing 3; Diskontieren eines Kapitals=Listing 4; Erstellung eines Tilgungsplans=Listing 5). Sie erkennen Ihre augenblickliche Position an der blinkenden Zahl. Nach beendeter Wahl drücken Sie »RETURN«. Dann wird das Programm geladen, dessen Zahl gerade blinkt und an der sich nach »RETURN« ein ← (Pfeil links) befindet.

Bedienungsanleitung für das Programm: »Erstellen eines Tilgungsplanes.«

Nach erfolgter Auswahl im Hauptmenü wird das Programm automatisch gestartet. Durch entsprechende Informationen in der Fußzeile erklärt sich das Programm im wesentlichen selbst. Allerdings mußte manchmal wegen Platzmangel darauf verzichtet werden.

Grundsätzlich gilt:

— Bis auf die Funktionstasten erfolgt bei jeder Eingabe und Auswahl ein Abschluß mit »RETURN«.

— Bei der Auswahl im Programm, wird immer mit den Tasten F3 und F5 gewählt, nach Drücken von »RETURN« wird immer das als nächstes ausgeführt, was zuletzt geblinkt hat.

— Zum Abschluß der Dateneingabe gilt für die Abfrage, ob die Eingabe korrekt ist, folgendes:

a) Zugelassen sind die Buchstaben j (ja) und n (nein).

b) Die Antwort kann bis zum Abschluß mit »RETURN« durch die Taste »DEL« korrigiert werden.

— Während der Dateneingabe kann man durch Drücken der Taste F1 zum ersten Auswahlmenü zurückkehren.

— Wird die Tabelle auf dem Bildschirm ausgedruckt, kann man den Bildschirmausschnitt innerhalb der festgesetzten Grenzen verändern.

Mit F1 kehrt man zur alten Bildschirmmaske zurück.

Das Ausgabefeld »Resultat« enthält folgende Informationen:

— Zinsaufwand (gesamt): Dieser Summe entsprechen sämtliche in den einzelnen Jahren zu zahlende Zinsen.

— Annuität (gesamt): Dies ist die Summe, die man insgesamt zurückzahlen muß (Zins- und Tilgungsanteil zusammen),

Im linken unteren Feld wird noch einmal angegeben, wie hoch der Zinsanteil von der Annuität in Prozent ist. Dieses Programm stellt Tilgungspläne bis zu einer Laufzeit von 50 Jahren auf. Falls dies nicht ausreicht, braucht in Zeile neun nur die Dimensionierung der Variablen e und e$ erhöht werden.

(Klaus Klöker/og)

	Programmdokumentation: Menü
200 — 260:	Aufbau der Bildschirmmaske
300 — 390:	Auswahlroutine durch Betätigen der Funktionstasten
400 — 410:	Sprung in die Zeile, in der das entsprechende Programm geladen, oder in der das Programm beendet wird
900 — 960:	Festlegung der Variablen

	Variablenliste: Menü
na$ (...) =	Nummern der Programme
nb$ (...) =	Namen der Programme
Z =	Zeile
S =	Spalte
Y =	Index der indizierten Variablen
g$ =	Eingabevariable der get-Anweisung

	Programmdokumentation: Zinsrechnung
10 — 30	Unterprogramm zum Löschen des rechten oberen und linken unteren Bildschirmfensters
70 — 75	Unterprogramm zur Rundung der Ergebnisse und zur Umwandlung in einen String
80	Unterprogramm zum Laden des Menüs
90	Unterprogramm zum Setzen des Cursors, an die Stelle, welche mit Z = Zeile und S = Spalte vor dem Sprungbefehl festgelegt wurde
95	Unterprogramm zum Zeichnen einer Linie
100 — 195	Auswahlroutine mit den Funktionstasten der entsprechenden reversen Felder oben links und unten links.
200 — 290	Auswahlroutine mit den Funktionstasten der Fußzeile.
300 — 480	Eingaberoutine
1000 — 1180	Beginn des Hauptprogramms: — Aufbau der Bildschirmmaske — Sprung in die Auswahlroutine, um festzulegen, was berechnet werden soll.
2000 — 2140	Aufbau der Eingabemaske und Eingabe der Daten mittels der Eingaberoutine. Anschließend erfolgt Abfrage, ob die Eingabe korrekt ist.
3000 — 3040	Berechnung des vorher ausgewählten Ergebnisses
3500 — 3610	Ausdruck des Ergebnisses. Es werden ausgedruckt: — die am Anfang festgelegten Kriterien — die eingegebenen Daten — das Endergebnis Es folgt: Aufbau der Fußzeile mit der Möglichkeit durch Wahl mit den Funktionstasten entsprechend zu verzweigen.

	Variablenliste: Zinsrechnung
ma$:	Kopfzeile
mb$, mc$, md$:	Fußzeilen
a$ (...):	Eingabekriterien
ka$; p$; zi$ ja$; mo$; ta$	Ein- und Ausgabekriterien
pm$; pu$:	Punkte in der Eingabemaske
Z:	Zeile der Cursorposition
S:	Spalte der Cursorposition
Z1:	niedrigste Zeile, die gelöscht werden soll
Z2:	höchste Zeile, die gelöscht werden soll
Y:	Variablenindex
Y1:	kleinster Index
Y2:	größter Index
SW:	Schrittweite
k; c:	Konstanten, die, je nach ihrer Wertigkeit, bestimmte Vorgaben ausführen.
f:	gibt an, welche Berechnung erfolgen soll
e:	gibt an, die Art der Laufzeit
h:	legt die Art der Laufzeit in der Berechnung fest
in$:	Variable in der Eingaberoutine
ab$:	Endergebnis

Programmdokumentation: Tilgungsrechnung

10 — 20	Unterroutine zum Löschen der rechten Bildschirmhälfte
30 — 55	Unterroutine zum Runden des Ergebnisses und zum Umwandeln in einen String
70	Unterroutine zum Laden des Menüs
80	Unterroutine zum Zeichen einer Linie
90	Unterroutine zur Festlegung der Cursorposition
95	Unterroutine zum Löschen der Fußzeile
100 — 195	Unterroutine zur Auswahl der vorgegebenen Möglichkeiten mit den Tasten F3 und F5, der Abschluß erfolgt mit »RETURN«.
200 — 290	Unterroutine zur Auswahl bei der Fußzeile mit den Tasten F3 und F5, der Abschluß erfolgt mit »RETURN«.
300 — 480	Eingaberoutine. Durch Drücken der Taste F1 kann man von vorne beginnen
900 — 960	Aufbau der Bildschirmmaske
1010 — 1080	Ausdruck der Kopfzeile und Bildschirmmaske, Sprung in die Auswahlroutine (100-195)
2000 — 2140	Aufbau der Eingabemaske und Sprung in die Eingaberoutine. Zum Schluß wird abgefragt, ob die Eingabe korrekt ist. Die Antwort wird mit »RETURN« eingegeben und kann vorher mit »DEL« korrigiert werden.
3000 — 3720	Ausführung der Rechnung mit Sprung in die Rundungs- und Stringumwandlungsroutine
3800 — 3900	Ausdruck des Resultates auf dem Bildschirm
3910 — 3950	Aufbau der Wahlfußzeile und Sprung in die Auswahlroutine (200-290)
4000 — 4590	Ausdruck der Tabelle und Verändern des Bildschirmausdrucks mit den Cursortasten: Mit F1 erfolgt ein Rücksprung
5000 — 5090	Änderung der eingegebenen Daten
5500 — 5660	Papierausdruck der eingegebenen Daten und der Tabelle
6000 — 6500	Festlegung der Variablen

Variablenliste: Tilgungsrechnung

ma$	Kopfzeile
mb$	
mc$	
me$	Fußzeilen
mf$	
mh$	
md$	Löschzeile zum Löschen des Bildschirms
Z:	Zeile der Cursorposition
S:	Spalte der Cursorposition
Y:	Variablenindex, der einfach dimensionierten Variablen
Y1:	niedrigster Variablenindex
Y2:	höchster Variablenindex
Sw:	Schriftweite
g$:	Variable der GET-Anweisung
g:	ASC (g$)
i,j:	Variablenindizes der zweifach dimensionierten Variablen
c:	Konstante, von der bestimmte Bedingungen abhängen
f:	gibt das Ergebnis der Auswahlroutine an
h:	beschränkt die maximale Eingabelänge
in$:	Variablenindex
pns:	füllt den Ergebnisstring mit Punkten auf
is:	gibt das Jahr in der Tabelle an
tb$:	chr$(16): Tabulator
ta$:	chr$(10): Leerzeile
X1:	Variable die den kleinsten Index der Tabelle angibt
X2:	Variable die den größten Index der Tabelle angibt
	Variablen der Rechnung t1,t2,J2,r1,a1,Z1,Z2

```
70 REM BY KLAUS KLOEKER                          <051>
80 GOTO 100                                      <008>
90 POKE 214,Z:POKE 211,S:SYS 58732:RETURN        <076>
100 REM * MENUE *                                <069>
110 PRINT"{CLR,WHITE}":POKE 53280,0:POKE 5
    3281,0:POKE 53272,23:POKE 788,52:GOSUB
     900                                         <225>
200 PRINT"{RVSON,4SPACEeE{SHIFT-SPACE}I{SH
    IFT-SPACE}N{SHIFT-SPACE}A{SHIFT-SPACE}
    N{SHIFT-SPACE}Z{SHIFT-SPACE}M{SHIFT-SP
    ACE}A{SHIFT-SPACE}I{SHIFT-SPACE}H{SHIF
    T-SPACE}E{SHIFT-SPACE}M{SHIFT-SPACE}A{
    SHIFT-SPACE}I I{SHIFT-SPACE}K{4SPACE,R
    VOFF}"                                       <098>
210 Z=4:S=6:GOSUB 90:PRINT"MENUE :":Z=5:GO
    SUB 90:PRINT"--------"                       <186>
220 Z=7:S=0:Y=1:FOR I=1 TO 5:GOSUB 90:PRIN
    T NA$(I);NB$(I)                              <153>
230 Z=Z+2:Y=Y+1:NEXT                             <249>
240 Z=18:GOSUB 90:PRINT"{RVSON,5SPACE,RVOF
    F,2SPACE}WAEHLEN SIE MIT{SPACE,RVSON,S
    PACE}F3{SPACE,RVOFF,SPACE}UND{SPACE,RV
    SON,SPACE}F5{SPACE,RVOFF,SPACE}!"            <077>
250 Z=20:S=7:GOSUB 90:PRINT"DRUECKEN SIE D
    ANN ' RETURN{SHIFT-SPACE}'{SPACE,SHIFT
    -SPACE}!"                                    <055>
260 Z=23:S=0:GOSUB 90:PRINT"TTTTTTTTTTTTTT
    TTTTTTTTTTTTTTTTTTTTTTTTTT"                  <176>
300 REM * AUSWAHLROUTINE *                       <101>
310 Z=7:Y=1                                      <208>
320 GOSUB 90:PRINT"{RVSON,5SPACE,RVOFF}":F
    OR I=1 TO 75:NEXT                            <246>
330 PRINT"{UP}";NA$(Y):FOR I=1 TO 125:NEXT <172>
340 GET G$:IF G$<>CHR$(134)AND G$<>CHR$(13
    5)AND G$<>CHR$(13)THEN 320                   <202>
350 IF G$=CHR$(13)THEN GOSUB 90:PRINT"{RVS
    ON}>{RVOFF}";:GOTO 400                       <165>
355 IF G$=CHR$(135)THEN 380                      <036>
360 Z=Z+2:Y=Y+1:IF Z>15 THEN Z=7:Y=1             <119>
370 GOTO 320                                     <092>
380 Z=Z-2:Y=Y-1:IF Z<7 THEN Z=15:Y=5             <006>
390 GOTO 320                                     <112>
400 ON Y GOTO 500,510,520,530,540                <091>
410 END                                          <158>
500 LOAD NB$(1),8                                <097>
510 LOAD NB$(2),8                                <236>
520 LOAD NB$(3),8                                <120>
530 LOAD NB$(4),8                                <003>
540 PRINT"{CLR}":POKE 53272,21:POKE 788,49
    :END                                         <071>
900 REM * VARIABLEN *                            <039>
910 NA$(1)="{RVSON}- 1 -{RVOFF,SPACE}":NA$
    (2)="{RVSON}- 2 -{RVOFF,SPACE}":NA$(3)
    ="{RVSON}- 3 -{RVOFF,SPACE}"                 <145>
920 NA$(4)="{RVSON}- 4 -{RVOFF,SPACE}":NA$
    (5)="{RVSON}- 5 -{RVOFF,SPACE}"              <035>
930 NB$(1)="ZINSRECHNUNG":NB$(2)="ZINSESZI
    NSRECHNUNG"                                  <148>
940 NB$(3)="DISKONTIEREN EINES KAPITALS"         <130>
950 NB$(4)="ERSTELLEN EINES TILGUNGSPLAENE
    S":NB$(5)="PROGRAMMENDE"                     <020>
960 RETURN                                       <000>
```

© 64'er

Listing 1. »Menüprogramm«.
Bitte beachten Sie die Eingabehinweise auf Seite 16.

```
7 POKE 45,PEEK(174):POKE 46,PEEK(175):CLR  <013>
8 REM BY KLAUS KLOEKER                     <245>
9 CLR:DIM A$(15),A(15):GOTO 1000           <249>
10 Z1=3:Z2=9:S=13:REM * MASKE LOESCHEN *   <217>
20 FOR Z=Z1 TO Z2:GOSUB 90:PRINT"{26SPACE}
   ":NEXT:IF C=0 THEN RETURN               <093>
30 Z1=12:Z2=20:C=0:GOTO 20                 <244>
50 RETURN                                  <108>
70 A$(12)=STR$(INT(A(12)*100+.5)):REM * RU
   NDEN                                    <101>
75 AB$=LEFT$(A$(12),LEN(A$(12))-2)+"."+RIG
   HT$(A$(12),2):RETURN                    <152>
80 LOAD"MENUE",8:REM *LOAD MENUE           <006>
```

Listing 2. Zinsrechnung

```
90 POKE 214,Z:POKE 211,S:SYS 58732:RETURN:
   REM * CURSORPOSITION                      <175>
95 A=55869:FOR I=1597 TO 1622:POKE I,100:P
   OKE A,1:A=A+1:NEXT:RETURN                 <110>
100 REM * AUSWAHLROUTINE *                   <155>
110 GOSUB 90:PRINT"{RVSON,11SPACE,RVOFF}":
    FOR I=1 TO 75:NEXT I                     <075>
120 GOSUB 90:PRINT"{RVSON}"A$(Y)"{RVOFF}":
    FOR I=1 TO 125:NEXT                      <112>
130 GET G$:IF G$<>CHR$(133)AND G$<>CHR$(13
    4)AND G$<>CHR$(135)AND G$<>CHR$(13)THE
    N 110                                    <023>
140 IF G$=CHR$(133)AND C THEN C=0:RETURN     <151>
150 IF G$=CHR$(13)THEN GOSUB 90:PRINT"{RVS
    ON}>{RVOFF}":RETURN                      <147>
160 IF G$=CHR$(135)THEN 190                  <095>
170 Z=Z+SW:Y=Y+1:IF Z>Z2 THEN Z=Z1:Y=Y1      <115>
180 GOTO 110                                 <124>
190 Z=Z-SW:Y=Y-1:IF Z<Z1 THEN Z=Z2:Y=Y2      <212>
195 GOTO 110                                 <139>
200 REM * WAHL-FUSSZEILE *:S=0               <128>
210 Z=23:GOSUB 90:PRINT"{RVSON,10SPACE,RVO
    FF}";:FOR I=1 TO 75:NEXT                 <218>
220 GOSUB 90:PRINT"{RVSON}"BA$(Y)"{RVOFF}"
    ;:FOR I=1 TO 175:NEXT                    <141>
230 GET G$:IF G$<>CHR$(134)AND G$<>CHR$(13
    5)AND G$<>CHR$(13)THEN 210               <217>
240 IF G$=CHR$(13)THEN GOSUB 90:PRINT"{RVS
    ON}>{RVOFF}";:RETURN                     <145>
250 IF G$=CHR$(135)THEN 280                  <153>
260 S=S+10:Y=Y+1:IF S>31 THEN S=0:Y=1        <107>
270 GOTO 210                                 <224>
280 S=S-10:Y=Y-1:IF S<0 THEN S=30:Y=4        <058>
290 GOTO 210                                 <244>
300 REM * EINGABEROUTINE *                   <179>
310 GOSUB 90                                 <058>
320 IN$=""                                   <010>
330 PRINT"T{LEFT}";                          <197>
340 GET G$:IF G$=""THEN 340                  <075>
350 G=ASC(G$):IF G=13 THEN GOTO 460          <165>
370 IF G$<"0"OR G$>"9"THEN 420               <081>
380 IN$=IN$+G$                               <221>
390 PRINT G$;                                <209>
400 IF LEN(IN$)>N THEN FOR I=1 TO LEN(IN$)
    :PRINT CHR$(20);:NEXT:GOTO 320           <227>
410 GOTO 330                                 <148>
420 IF G=46 THEN 380                         <130>
425 IF G$=CHR$(133)THEN 9                    <253>
430 IF G<>20 THEN 330                        <135>
440 IF LEN(IN$)<1 THEN 330                   <140>
450 IN$=LEFT$(IN$,LEN(IN$)-1):GOTO 390       <227>
460 PN$="":FOR I=1 TO(10-LEN(IN$)):PN$=PN$
    +".":NEXT:PRINT PN$                      <017>
470 IF LEN(IN$)<1 THEN 310                   <106>
480 RETURN                                   <028>
999 :                                        <213>
1000 REM ** ZINSRECHNUNG **                  <021>
1010 GOSUB 5000:PRINT MA$                    <098>
1020 A=55388:FOR I=1076 TO 1906 STEP 40:PO
     KE I,101:POKE A,1:A=A+40:NEXT           <078>
1030 A=55736:FOR I=1464 TO 1474:POKE I,100
     :POKE A,1:A=A+1:NEXT                    <183>
1040 A=55749:FOR I=1477 TO 1502:POKE I,100
     :POKE A,1:A=A+1:NEXT                    <032>
1050 Z=23:S=0:GOSUB 90:PRINT MB$             <154>
1060 Z=3:GOSUB 90:PRINT"AUSWAHL :":PRINT"T
     TTTTTTTT"                               <153>
1070 Z=5:FOR I=1 TO 5:GOSUB 90:PRINT"{RVSO
     N}"A$(I)"{RVOFF}":Z=Z+1:NEXT            <066>
1080 Z=5:Z1=Z:Z2=9:S=0:Y=1:Y1=Y:Y2=5:SW=1:
     C=0:GOSUB 100:F=Y:IF F=5 THEN 80        <255>
1090 Z=13:S=0:GOSUB 90:PRINT"LAUFZEIT IN":
     PRINT"TTTTTTTTTTTT"                     <044>
1100 Z=15:FOR I=6TO 8:S=0:GOSUB 90:PRINT"
     {RVSON}"A$(I)"{RVOFF}":Z=Z+1:NEXT       <104>
1110 Z=15:Z1=Z:Z2=17:S=0:Y=6:Y1=Y:Y2=8:SW=
     1:C=0:GOSUB 100:E=Y                     <206>
1120 IF E=6 THEN A$(11)=JA$:H=1              <118>
1130 IF E=7 THEN A$(11)=MO$:H=12             <174>
1140 IF E=8 THEN A$(11)=TA$:H=360            <064>
1150 IF F=1 THEN A$(9)=KA$:A$(10)=P$:K=1     <203>
1160 IF F=2 THEN A$(9)=ZI$:A$(10)=P$:K=1     <217>
1170 IF F=3 THEN A$(9)=KA$:A$(10)=ZI$:K=0    <087>
1180 IF F=4 THEN A$(9)=KA$:A$(10)=ZI$:A$(1
     1)=P$:K=2                               <090>
2000 REM * EINGABEMASKE *                    <128>
2010 Z=3:S=13:GOSUB 90:PRINT"EINGABE :":Z=
     4:GOSUB 90:PRINT"TTTTTTTTT"             <033>
2020 Z=5:FOR I=9 TO 11:S=13:GOSUB 90:PRINT
     "{RVSON}"A$(I)"{RVOFF,SPACE}: ";PM$     <153>
2030 Z=Z+2:NEXT:Z=23:S=0:GOSUB 90:PRINT MC
     $                                       <013>
2040 Z=5:S=27:N=10:GOSUB 300:A(9)=VAL(IN$)   <196>
2050 Z=7:S=27:N=10:GOSUB 300:A(10)=VAL(IN$
     )                                       <255>
2060 Z=9:S=27:N=10:GOSUB 300:A(11)=VAL(IN$
     )                                       <042>
2070 Z=13:S=13:GOSUB 90:PRINT"EINGABE KORR
     EKT ? (J/N):":GOSUB 95                  <086>
2080 Z=13:S=38:GOSUB 90:PRINT"T{LEFT}";      <065>
2090 GET G$:IF G$<>"J"AND G$<>"N"THEN 2090   <206>
2100 PRINT G$:IF G$="J"THEN Y=1              <075>
2110 IF G$="N"THEN Y=0                       <107>
2120 GET I$:IF I$<>CHR$(13)AND I$<>CHR$(20
     )THEN 2120                              <136>
2130 IF I$=CHR$(20)THEN 2080                 <057>
2140 C=1:GOSUB 10:IF Y=0 THEN 2010           <132>
3000 REM * RECHNUNG *                        <043>
3010 IF F=1 THEN A(12)=(A(9)*A(10)*A(11))/
     (100*H)                                 <103>
3020 IF F=2 THEN A(12)=(A(9)*100*H)/(A(10)
     *A(11))                                 <216>
3030 IF F=3 THEN A(12)=(A(10)*100*H)/(A(9)
     *A(11))                                 <015>
3040 IF F=4 THEN A(12)=(A(10)*100*H)/(A(9)
     *A(11))                                 <057>
3500 REM * BILDSCHIRM                        <233>
3510 Z=3:S=13:GOSUB 90:PRINT"EINGABE :":Z=
     4:GOSUB 90:PRINT"TTTTTTTTTT"            <122>
3520 Z=5:FOR I=9 TO 11:GOSUB 90:PRINT A$(I
     )" :"A(I):Z=Z+2:NEXT                    <016>
3530 Z=13:GOSUB 90:PRINT"RESULTAT :":Z=14:
     GOSUB 90:PRINT"TTTTTTTTTTT":GOSUB 70    <064>
3540 Z=16:GOSUB 90:PRINT"{RVSON}"A$(F)"{RV
     OFF,SPACE}:";:PRINT TAB(38-LEN(AB$));
     AB$                                     <149>
3550 PRINT TAB(39-LEN(AB$));:FOR I=1 TO LE
     N(AB$)-1:PRINT"=";:NEXT                 <170>
3560 Z=20:GOSUB 90:IF K=1 THEN PRINT"AUSDR
     UCK IN DM{SHIFT-SPACE}!":GOTO 3580      <244>
3570 IF K=0 THEN PRINT"AUSDRUCK IN PROZENT
     !"                                      <255>
3580 Z=23:S=0:GOSUB 90:FOR I=1 TO 4:PRINT"
     {RVSON}"BA$(I)"{RVOFF}";:NEXT           <234>
3590 Y=1:GOSUB 200:IF Y=4 THEN 80            <041>
3600 IF Y=3 THEN 1010                        <136>
3610 IF Y=2 THEN 4500                        <247>
4000 REM * AENDERUNG *                       <039>
4010 C=1:GOSUB 10                            <239>
4020 Z=3:S=13:GOSUB 90:PRINT"AENDERUNG :":
     Z=4:GOSUB 90:PRINT"TTTTTTTTTTTT"        <139>
4030 Z=5:S=13:FOR I=9 TO 11:GOSUB 90:PRINT
     "{RVSON}"A$(I)"{RVOFF,SPACE}:";A(I):Z
     =Z+2:NEXT                               <030>
4040 Z=23:S=0:GOSUB 90:PRINT MD$;            <059>
4050 Z=5:Z1=Z:Z2=9:S=13:Y=9:Y1=Y:Y2=11:SW=
     2:C=1:GOSUB 100                         <253>
4060 IF C=0 THEN GOSUB 10:GOTO 3000          <068>
4070 Z=13:S=13:GOSUB 90:PRINT"{RVSON}"A$(Y
     )"{RVOFF,SPACE}: ";PM$:GOSUB 95         <196>
4080 Z=23:S=0:GOSUB 90:PRINT MC$;:Z=13:S=2
     7:GOSUB 300:A(Y)=VAL(IN$):GOTO 4010     <211>
4500 REM * AUSDRUCK *                        <254>
4510 OPEN 4,4,7:PRINT#4,CHR$(14);CHR$(16);
     "28ZINSRECHNUNG"                        <058>
4520 PRINT#4,CHR$(16);"28-------------";CHR
     $(10)                                   <171>
4530 FOR I=9 TO 11:PRINT#4,CHR$(15);CHR$(1
     6);"10*** "A$(I)" : ";A(I):NEXT         <003>
4540 GOSUB 70:PRINT#4,CHR$(10);CHR$(16);"1
     4"A$(F)" : "AB$                         <141>
4550 PRINT#4,CHR$(16);"15";:FOR I=1 TO 2+L
     EN(A$(F)+AB$):PRINT#4,"=";:NEXT         <178>
4560 PRINT#4:PRINT#4:PRINT#4                 <213>
4570 PRINT#4,CHR$(13);CHR$(16);"10***{3SPA
     CE}KAPITAL UND ZINSSATZ WERDEN";        <209>
4580 PRINT#4," IN ' DM ' AUSGEDRUCKT,"       <135>
4590 PRINT#4,CHR$(16);"16DER ZINSSATZ IN '
```

Listing 2. Zinsrechnung (Fortsetzung)

```
     % ' !"                                    <036>
4600 PRINT#4:FOR I=1 TO 40:PRINT#4,"==";:N
     EXT:PRINT#4:PRINT#4:PRINT#4:PRINT#4       <072>
4610 CLOSE 4:FOR I=56176 TO 56215:POKE I,0
     :NEXT:GOTO 3580                           <180>
5000 REM * VARIABLENLISTE *                    <016>
5010 MA$="{CLR,RVSON,8SPACE}Z{SHIFT-SPACE}
     I{SHIFT-SPACE}N{SHIFT-SPACE}S{SHIFT-S
     PACE}R{SHIFT-SPACE}E{SHIFT-SPACE}C{SH
     IFT-SPACE}H{SHIFT-SPACE}N{SHIFT-SPACE
     }U{SHIFT-SPACE}N{SHIFT-SPACE}G{8SPACE
     ,RVOFF}"                                  <159>
5020 MB$="{RVSON,SPACE}F3 U. F5 - WAHL ! '
     RETURN' - EINGABE !{RVOFF}"               <247>
5030 MC$="{RVSON,3SPACE}F1 - NEU !{5SPACE}
     GEBEN SIE EIN !{6SPACE,RVOFF}"            <135>
5040 MD$="{RVSON,SPACE}F1 - RESULTAT !{5SP
     ACE}F3 U. F5 - WAHL !{SPACE,RVOFF}"       <167>
5050 KA$="{2SPACE}KAPITAL{2SPACE}":P$="{2S
     PACE}ZINSSATZ ":ZI$=" ZINSERTRAG"         <229>
5060 JA$="{2SPACE}JAHRE{4SPACE}":MO$="{2SP
     ACE}MONATE{3SPACE}":TA$="{2SPACE}TAGE
     {5SPACE}":PM$="..........."               <203>
5110 BA$(1)=" AENDERN{2SPACE}"                 <245>
5120 BA$(2)=" KOPIE{4SPACE}"                   <209>
5130 BA$(3)=" N.BERECH."                       <052>
5140 BA$(4)=" MENUE{4SPACE}"                   <126>
5510 A$(1)=" ZINSERTRAG"                       <044>
5520 A$(2)="{2SPACE}KAPITAL{2SPACE}"           <065>
5530 A$(3)="{2SPACE}ZINSSATZ "                 <154>
5540 A$(4)="{2SPACE}LAUFZEIT "                 <069>
5550 A$(5)="{2SPACE}MENUE{4SPACE}"             <146>
5560 A$(6)="{2SPACE}JAHRE{4SPACE}"             <070>
5570 A$(7)="{2SPACE}MONATE{3SPACE}"            <137>
5580 A$(8)="{2SPACE}TAGE{5SPACE}"              <234>
5590 A$(13)="{2SPACE}AENDERN{2SPACE}"          <205>
5600 A$(14)="{2SPACE}RESULTAT "                <113>
5610 A$(15)="{2SPACE}MENUE{4SPACE}":RETURN     <097>
```

© 64'er

Listing 2. Zinsrechnung (Schluß)

```
7 POKE 45,PEEK(174):POKE 46,PEEK(175):CLR      <013>
9 CLR:DIM A$(19),A(19),BA$(4):GOTO 1000        <185>
10 Z1=2:Z2=10:S=13:REM * MASKELOESCHEN *       <055>
20 FOR Z=Z1 TO Z2:GOSUB 80:PRINT"{26SPACE}
   ":NEXT:IF C=0 THEN RETURN                   <077>
30 Z1=12:Z2=20:C=0:GOTO 20                     <244>
40 S=0:FOR Z=12 TO 20:GOSUB 80:PRINT"{11SP
   ACE}":NEXT:RETURN                           <055>
50 A$(A)=STR$(INT(A(A)*100+.5)):REM * RUND
   EN                                          <253>
60 AB$=LEFT$(A$(A),LEN(A$(A))-2)+"."+RIGHT
   $(A$(A),2):RETURN                           <161>
70 LOAD"FM.MENUE",8:REM *                      <248>
80 POKE 214,Z:POKE 211,S:SYS 58732:RETURN:
   REM * CURSORPOSITION                        <165>
90 A=55869:FOR I=1597 TO 1622:POKE I,100:P
   OKE A,1:A=A+1:NEXT:RETURN                   <105>
100 REM * AUSWAHLROUTINE                       <243>
110 GOSUB 80:PRINT"{RVSON,11SPACE,RVOFF}":
    FOR I=1 TO 75:NEXT                         <026>
120 GOSUB 80:PRINT"{RVSON}"A$(Y)"{RVOFF}":
    FOR I=1 TO 125:NEXT                        <104>
130 GET G$:IF G$<>CHR$(133)AND G$<>CHR$(13
    4)AND G$<>CHR$(135)AND G$<>CHR$(13)THE
    N 110                                      <023>
140 IF G$=CHR$(133)AND C THEN C=0:RETURN       <151>
150 IF G$=CHR$(13)THEN GOSUB 80:PRINT"{RVS
    ON}>{RVOFF}";:RETURN                       <023>
160 IF G$=CHR$(135)THEN 190                    <095>
170 Z=Z+SW:Y=Y+1:IF Z>Z2 THEN Z=Z1:Y=Y1        <115>
180 GOTO 110                                   <124>
190 Z=Z-SW:Y=Y-1:IF Z<Z1 THEN Z=Z2:Y=Y2        <212>
195 GOTO 110                                   <139>
200 REM * WAHL-FUSSZEILE:S=0                   <122>
210 Z=23:GOSUB 80:PRINT"{RVSON,10SPACE,RVO
    FF}";:FOR I=1 TO 75:NEXT                   <217>
220 GOSUB 80:PRINT"{RVSON}"BA$(Y)"{RVOFF}"
    ;:FOR I=1 TO 175:NEXT                      <133>
230 GET G$:IF G$<>CHR$(134)AND G$<>CHR$(13
    5)AND G$<>CHR$(13)THEN 210                 <217>
240 IF G$=CHR$(13)THEN GOSUB 80:PRINT"{RVS
    ON}>{RVOFF}";:RETURN                       <113>
250 IF G$=CHR$(135)THEN 280                    <153>
260 S=S+10:Y=Y+1:IF S>30 THEN S=0:Y=1          <099>
270 GOTO 210                                   <224>
280 S=S-10:Y=Y-1:IF S<0 THEN S=30:Y=4          <058>
290 GOTO 210                                   <244>
300 REM * EINGABEROUTINE                       <011>
310 GOSUB 80                                   <050>
315 IN$=""                                     <005>
320 PRINT"T{LEFT}";                            <187>
330 GET G$:IF G$=""THEN 330                    <000>
340 G=ASC(G$):IF G=13 THEN 460                 <177>
360 IF G<48 OR G>57 THEN 410                   <148>
370 IN$=IN$+G$:IN=LEN(IN$)                     <156>
380 PRINT G$;                                  <199>
390 IF IN>N THEN FOR I=1 TO IN:PRINT CHR$(
    20);:NEXT:GOTO 310                         <067>
400 GOTO 320                                   <122>
410 IF G=46 THEN 370                           <118>
420 IF G$=CHR$(133)THEN 9                      <248>
430 IF G<>20 THEN 320                          <131>
440 IF IN<1 THEN 320                           <002>
450 IN$=LEFT$(IN$,IN-1):GOTO 380               <211>
460 PN$="":FOR I=1 TO(10-LEN(IN$)):PN$=PN$
    +".":NEXT:PRINT PN$                        <017>
470 IF LEN(IN$)<1 THEN 310                     <106>
480 RETURN                                     <028>
999 :                                          <213>
1000 REM * ZINSESZINSRECHNUNG                  <043>
1010 GOSUB 5000:PRINT MA$                      <098>
1020 A=55388:FOR I=1076 TO 1906 STEP 40:PO
     KE I,101:POKE A,1:A=A+40:NEXT             <078>
1030 A=55736:FOR I=1464 TO 1474:POKE I,100
     :POKE A,1:A=A+1:NEXT                      <183>
1040 A=55749:FOR I=1477 TO 1502:POKE I,100
     :POKE A,1:A=A+1:NEXT                      <032>
1050 Z=23:S=0:GOSUB 80:PRINT MB$               <138>
1060 Z=2:GOSUB 80:PRINT"AUSWAHL :":PRINT"T
     TTTTTTTT"                                 <009>
1070 Z=4:FOR I=1 TO 5:GOSUB 80:PRINT"{RVSO
     N}"A$(I)"{RVOFF}":Z=Z+1:NEXT              <242>
1080 Z=4:Z1=Z:Z2=8:S=0:Y=1:Y1=Y:Y2=5:SW=1:
     C=0:GOSUB 100:F=Y:IF F=5 THEN 70          <173>
1090 IF F=1 THEN A$(9)=KO$:A$(10)=P$:A$(11
     )=N$:K=1                                  <212>
1100 IF F=2 THEN A$(9)=KN$:A$(10)=P$:A$(11
     )=N$:K=1                                  <190>
1110 IF F=3 THEN A$(9)=KO$:A$(10)=KN$:A$(1
     1)=N$:K=0                                 <096>
1120 IF F=4 THEN A$(9)=KO$:A$(10)=KN$:A$(1
     1)=P$:K=2                                 <156>
2000 REM * EINGABEMASKE                        <086>
2010 Z=2:S=13:GOSUB 80:PRINT"EINGABE :":Z=
     3:GOSUB 80:PRINT"TTTTTTTTT"               <224>
2020 Z=4:FOR I=9 TO 12:S=13:GOSUB 80:PRINT
     "{RVSON}"A$(I)"{RVOFF,SPACE}: ";PM$:Z
     =Z+2:NEXT                                 <210>
2030 Z=23:S=0:GOSUB 80:PRINT MC$;              <063>
2040 Z=4:S=27:N=10:GOSUB 300:A(9)=VAL(IN$)     <180>
2050 Z=6:GOSUB 300:A(10)=VAL(IN$)              <089>
2060 Z=8:GOSUB 300:A(11)=VAL(IN$)              <195>
2065 Z=10:N=9:GOSUB 300:A(12)=VAL(IN$)         <216>
2070 Z=13:S=13:GOSUB 80:PRINT"EINGABE KORR
     EKT ? (J/N):":GOSUB 90                    <014>
2080 Z=13:S=38:GOSUB 80:PRINT"T{LEFT}";        <049>
2090 GET G$:IF G$<>"J"AND G$<>"N"THEN 2090     <206>
2100 PRINT G$:IF G$="J"THEN Y=1                <075>
2110 IF G$="N"THEN Y=0                         <107>
2120 GET I$:IF I$<>CHR$(13)AND I$<>CHR$(20
     )THEN 2120                                <136>
2130 IF I$=CHR$(20)THEN 2080                   <057>
2140 C=1:GOSUB 10:IF Y=0 THEN 2010             <132>
3000 REM * RECHNUNG                            <137>
3010 IF F=1 THEN A(16)=A(9)*(1+(A(10)/(A(1
     2)*100)))↑(A(11)*A(12))                   <169>
3020 IF F=1 THEN A(19)=100*((1+(A(10)/(A(1
     2)*100)))↑A(12)-1):D=1                    <054>
```

Listing 3. Zinseszinsrechnung

```
3030 IF F=2 THEN A(16)=A(9)/(1+(A(10)/(A(1
     2)*100)))↑(A(11)*A(12)):D=0            <101>
3040 IF F=3 THEN A(16)=((A(10)/A(9))↑(1/(A
     (11)*A(12)))-1)*100*A(12):D=0          <174>
3050 IF F=4 THEN A(16)=(LOG(A(10)/A(9))/LO
     G(1+(A(11)/(A(12)*100))))/A(12):D=0    <011>
3500 REM * SCREEN                           <009>
3510 Z=2:S=13:GOSUB 80:PRINT"EINGABE :":Z=
     3:GOSUB 80:PRINT"TTTTTTTTTT"           <057>
3520 Z=4:FOR I=9 TO 12:GOSUB 80:PRINT A$(I
     )" :"A(I):Z=Z+2:NEXT                   <144>
3530 Z=13:GOSUB 80::PRINT"RESULTAT :":Z=14
     :GOSUB 80:PRINT"TTTTTTTTTT":A=16:GOSU
     B 50                                   <200>
3540 Z=15:GOSUB 80:PRINT"{RVSON}"A$(F)"{RV
     OFF,SPACE}:"TAB(38-LEN(AB$));AB$       <010>
3550 PRINT TAB(39-LEN(AB$));:FOR I=1 TO LE
     N(AB$)-1:PRINT"=";:NEXT                <170>
3560 Z=20:GOSUB 80:IF K=1 THEN PRINT"AUSDR
     UCK IN DM !":GOTO 3580                 <036>
3570 IF K=0 THEN PRINT"AUSDRUCK IN PROZENT
      !"                                    <255>
3580 IF D=0 THEN 3620                       <115>
3590 Z=13:S=0:GOSUB 80::PRINT"{RVSON}"A$(1
     7)"{RVOFF}":PRINT"{RVSON}"A$(18)"{RVO
     FF}"                                   <244>
3600 Z=17:S=1:GOSUB 80:A=19:GOSUB 50:PRINT
      AB$" %"                               <124>
3610 PRINT TAB(7-LEN(AB$));:FOR I=1 TO LEN
     (AB$)+1:PRINT"=";:NEXT                 <012>
3620 Z=23:S=0:GOSUB 80:FOR I=1 TO 4:PRINT"
     {RVSON}"BA$(I)"{RVOFF}";:NEXT          <004>
3630 Y=1:GOSUB 200:IF Y=4 THEN 70           <080>
3640 IF Y=3 THEN 1010                       <176>
3650 IF Y=2 THEN 4500                       <031>
4000 REM * AENDERUNG                        <226>
4010 C=1:GOSUB 10:IF D THEN GOSUB 40        <003>
4020 Z=2:S=13:GOSUB 80::PRINT"AENDERUNG :"
     :Z=3:GOSUB 80:PRINT"TTTTTTTTTTT"       <149>
4030 Z=4:S=13::FOR I=9 TO 12:GOSUB 80:PRIN
     T"{RVSON}"A$(I)"{RVOFF,SPACE}:";A(I):
     Z=Z+2:NEXT                             <207>
4040 Z=23:S=0:GOSUB 80:PRINT MD$;           <043>
4050 Z=4:Z1=Z:Z2=10:S=13:Y=9:Y1=Y:Y2=12:SW
     =2:C=1:GOSUB 100                       <126>
4060 IF C=0 THEN GOSUB 10:GOTO 3000         <068>
4070 Z=13:S=13:GOSUB 80:PRINT"{RVSON}"A$(Y
     )"{RVOFF,SPACE}: ";PM$:GOSUB 90        <084>
4080 Z=23:S=0:GOSUB 80:PRINT MC$;:Z=13:S=2
     7:GOSUB 300:A(Y)=VAL(IN$):GOTO 4010    <195>
4500 REM * PRINTER                          <052>
4505 TA$=CHR$(10):TB$=CHR$(16)              <198>
4510 OPEN 4,4,7:PRINT#4,CHR$(14);TB$"25ZIN
     SESZINSRECHNUNG"                       <121>
4520 PRINT#4,TB$"25-----------------"TA$    <132>
4530 FOR I=9 TO 12:PRINT#4,CHR$(15)TB$"10*
     ** "A$(I)" : "A(I):NEXT                <049>
4540 A=16:GOSUB 50:PRINT#4,TA$;TB$"14"A$(F
     )" : "AB$                              <189>
4550 PRINT#4,TB$"15";:FOR I=1 TO 2+LEN(A$(
     F)+AB$):PRINT#4,"=";:NEXT              <076>
4560 PRINT#4,CHR$(13):IF D THEN GOSUB 4620 <041>
4565 PRINT#4,TA$;TA$;TA$                    <080>
4570 PRINT#4,CHR$(13);TB$"10***{2SPACE}KAP
     ITAL UND ZINSSATZ WERDEN";             <002>
4580 PRINT#4," IN ' DM ' AUSGEDRUCKT,"      <135>
4590 PRINT#4,TB$"16ZINSSATZ UND JAHRESZINS
      (FALLS VORHANDEN) IN ' %{SHIFT-SPACE
     }' !"                                  <078>
4600 PRINT#4:FOR I=1 TO 40:PRINT#4,"==";:N
     EXT:PRINT#4,TA$;TA$;TA$                <040>
4610 CLOSE 4:FOR I=56176 TO 56215:POKE I,0
     :NEXT:GOTO 3620                        <082>
4620 IF D THEN PRINT#4,TA$;TB$"15"A$(17)"
     "A$(18)" : ";                          <056>
4630 A=19:GOSUB 50:PRINT#4,AB$              <000>
4640 PRINT#4,TB$"15";:FOR I=1 TO LEN(A$(18
     )+A$(19)+AB$)+10:PRINT#4,"=";:NEXT     <010>
4650 RETURN                                 <136>
5000 REM * VARIABLEN                        <006>
5010 MA$="{CLR,RVSON,2SPACE}Z{SHIFT-SPACE}
     I{SHIFT-SPACE}N{SHIFT-SPACE}S{SHIFT-S
     PACE}E{SHIFT-SPACE}S{SHIFT-SPACE}Z{SH
     IFT-SPACE}I{SHIFT-SPACE}N{SHIFT-SPACE
     }S{SHIFT-SPACE}R{SHIFT-SPACE}E{SHIFT-
     SPACE}C{SHIFT-SPACE}H{SHIFT-SPACE}N{S
     HIFT-SPACE}U{SHIFT-SPACE}N{SHIFT-SPAC
     E}G{2SHIFT-SPACE,RVOFF}"               <166>
5020 MB$="{RVSON,SPACE}F3 U. F5 - WAHL ! '
     RETURN' - EINGABE !{RVOFF}"            <247>
5030 MC$="{RVSON,2SPACE}F1 - NEU !{6SPACE}
     GEBEN SIE EIN !{6SPACE,RVOFF}"         <135>
5040 MD$="{RVSON,SPACE}F1 - RESULTAT !{5SP
     ACE}F3 U. F5 - WAHL !{SPACE,RVOFF}"    <167>
5050 KO$=" AN.KAPITAL":P$="{2SPACE}ZINSSAT
     Z ":KN$=" ENDKAPITAL"                  <043>
5060 N$="{2SPACE}JAHRE{4SPACE}":A$(12)=" Z
     INSPER./A":PM$="..........."           <228>
5100 BA$(1)="{2SPACE}AENDERN "              <235>
5110 BA$(2)="{2SPACE}KOPIE{3SPACE}"         <197>
5120 BA$(3)="{2SPACE}NEU{5SPACE}"           <214>
5130 BA$(4)=" MENUE{4SPACE}"                <116>
5500 A$(1)=KN$                              <142>
5510 A$(2)=KO$                              <186>
5520 A$(3)=P$                               <003>
5530 A$(4)="{2SPACE}LAUFZEIT "              <059>
5540 A$(5)="{2SPACE}MENUE{4SPACE}"          <136>
5600 A$(13)="{2SPACE}AENDERN{2SPACE}"       <215>
5610 A$(14)="{2SPACE}RESULTAT "             <123>
5620 A$(15)="{2SPACE}MENUE{4SPACE}"         <020>
5630 A$(17)="EFFEKTIVER"                    <092>
5640 A$(18)="JAHRESZINS":RETURN             <098>
```


Listing 3. Zinseszinsrechnung (Schluß)

```
8 POKE 45,PEEK(174):POKE 46,PEEK(175):CLR  <014>
9 CLR:DIM A$(15),A(15):GOTO 1000           <249>
10 Z1=3:Z2=9:S=13:REM * MASKE LOESCHEN *   <217>
20 FOR Z=Z1 TO Z2:GOSUB 90:PRINT"{26SPACE}
   ":NEXT:IF C=0 THEN RETURN               <093>
30 Z1=12:Z2=20:C=0:GOTO 20                 <244>
50 RETURN                                  <108>
70 A$(12)=STR$(INT(A(12)*100+.5)):REM * RU
   NDEN                                    <101>
75 AB$=LEFT$(A$(12),LEN(A$(12))-2)+"."+RIG
   HT$(A$(12),2):RETURN                    <152>
80 LOAD"MENUE",8:REM *LOAD MENUE           <006>
90 POKE 214,Z:POKE 211,S:SYS 58732:RETURN:
   REM * CURSORPOSITION                    <175>
95 A=55869:FOR I=1597 TO 1622:POKE I,100:P
   OKE A,1:A=A+1:NEXT:RETURN               <110>
100 REM * AUSWAHLROUTINE *                 <155>
110 GOSUB 90:PRINT"{RVSON,11SPACE,RVOFF}":
    FOR I=1 TO 75:NEXT I                   <075>
120 GOSUB 90:PRINT"{RVSON}"A$(Y)"{RVOFF}":
    FOR I=1 TO 125:NEXT                    <112>
130 GET G$:IF G$<>CHR$(133)AND G$<>CHR$(13
    4)AND G$<>CHR$(135)AND G$<>CHR$(13)THE
    N 110                                  <023>
140 IF G$=CHR$(133)AND C THEN C=0:RETURN   <151>
150 IF G$=CHR$(13)THEN GOSUB 90:PRINT"{RVS
    ON}>{RVOFF}":RETURN                    <147>
160 IF G$=CHR$(135)THEN 190                <095>
170 Z=Z+SW:Y=Y+1:IF Z>Z2 THEN Z=Z1:Y=Y1    <115>
180 GOTO 110                               <124>
190 Z=Z-SW:Y=Y-1:IF Z<Z1 THEN Z=Z2:Y=Y2    <212>
195 GOTO 110                               <139>
200 REM * WAHL-FUSSZEILE *:S=0             <128>
210 Z=23:GOSUB 90:PRINT"{RVSON,10SPACE,RVO
    FF}";:FOR I=1 TO 75:NEXT               <218>
220 GOSUB 90:PRINT"{RVSON}"BA$(Y)"{RVOFF}"
    ;:FOR I=1 TO 175:NEXT                  <141>
230 GET G$:IF G$<>CHR$(134)AND G$<>CHR$(13
    5)AND G$<>CHR$(13)THEN 210             <217>
240 IF G$=CHR$(13)THEN GOSUB 90:PRINT"{RVS
    ON}>{RVOFF}";:RETURN                   <145>
250 IF G$=CHR$(135)THEN 280                <153>
```

Listing 4. Diskontieren eines Kapitals

```
260 S=S+10:Y=Y+1:IF S>31 THEN S=0:Y=1             <107>
270 GOTO 210                                      <224>
280 S=S-10:Y=Y-1:IF S<0 THEN S=30:Y=4             <058>
290 GOTO 210                                      <244>
300 REM * EINGABEROUTINE *                        <179>
310 GOSUB 90                                      <058>
320 IN$=""                                        <010>
330 PRINT"T{LEFT}";                               <197>
340 GET G$:IF G$=""THEN 340                       <075>
350 G=ASC(G$):IF G=13 THEN 460                    <187>
360 IF G=13 THEN RETURN                           <080>
370 IF G$<"0"OR G$>"9"THEN 420                    <011>
380 IN$=IN$+G$                                    <221>
390 PRINT G$;                                     <209>
400 IF LEN(IN$)>N THEN FOR I=1 TO LEN(IN$)
    :PRINT CHR$(20);:NEXT:GOTO 320                <227>
410 GOTO 330                                      <148>
420 IF G=46 THEN 380                              <130>
425 IF G$=CHR$(133)THEN 9                         <253>
430 IF G<>20 THEN 330                             <135>
440 IF LEN(IN$)<1 THEN 330                        <140>
450 IN$=LEFT$(IN$,LEN(IN$)-1):GOTO 390            <227>
460 PN$="":FOR I=1 TO(10-LEN(IN$)):PN$=PN$
    +".":NEXT:PRINT PN$                           <017>
470 IF LEN(IN$)<1 THEN 310                        <106>
480 RETURN                                        <028>
999 :                                             <213>
1000 REM ** DIKONTIEREN **                        <184>
1010 GOSUB 5000:PRINT MA$                         <098>
1020 A=55388:FOR I=1076 TO 1906 STEP 40:PO
     KE I,101:POKE A,1:A=A+40:NEXT                <078>
1030 A=55736:FOR I=1464 TO 1474:POKE I,100
     :POKE A,1:A=A+1:NEXT                         <183>
1040 A=55749:FOR I=1477 TO 1502:POKE I,100
     :POKE A,1:A=A+1:NEXT                         <032>
1050 Z=23:S=0:GOSUB 90:PRINT MB$                  <154>
1060 Z=3:GOSUB 90:PRINT"AUSWAHL :":PRINT"T
     TTTTTTTT"                                    <153>
1070 Z=5:FOR I=1 TO 5:GOSUB 90:PRINT"{RVSO
     N}"A$(I)"{RVOFF}":Z=Z+1:NEXT                 <066>
1080 Z=5:Z1=Z:Z2=9:S=0:Y=1:Y1=Y:Y2=5:SW=1:
     C=0:GOSUB 100:F=Y:IF F=5 THEN 80             <255>
1150 IF F=1 THEN A$(9)=KN$:A$(10)=P$:A$(11
     )=JA$:K=1                                    <062>
1160 IF F=2 THEN A$(9)=KB$:A$(10)=P$:A$(11
     )=JA$:K=1                                    <101>
1170 IF F=3 THEN A$(9)=KN$:A$(10)=KB$:A$(1
     1)=JA$:K=0                                   <029>
1180 IF F=4 THEN A$(9)=KN$:A$(10)=KB$:A$(1
     1)=P$:K=2                                    <128>
2000 REM * EINGABEMASKE *                         <128>
2010 Z=3:S=13:GOSUB 90:PRINT"EINGABE :":Z=
     4:GOSUB 90:PRINT"TTTTTTTTT"                  <033>
2020 Z=5:FOR I=9 TO 11:S=13:GOSUB 90:PRINT
     "{RVSON}"A$(I)"{RVOFF,SPACE}: ";PM$          <153>
2030 Z=Z+2:NEXT:Z=23:S=0:GOSUB 90:PRINT MC
     $                                            <013>
2040 Z=5:S=27:N=10:GOSUB 300:A(9)=VAL(IN$)        <196>
2050 Z=7:GOSUB 300:A(10)=VAL(IN$)                 <105>
2060 Z=9:GOSUB 300:A(11)=VAL(IN$)                 <211>
2070 Z=13:S=13:GOSUB 90:PRINT"EINGABE KORR
     EKT ? (J/N):":GOSUB 95                       <086>
2080 Z=13:S=38:GOSUB 90:PRINT"T{LEFT}";           <065>
2090 GET G$:IF G$<>"J"AND G$<>"N"THEN 2090        <206>
2100 PRINT G$:IF G$="J"THEN Y=1                   <075>
2110 IF G$="N"THEN Y=0                            <107>
2120 GET I$:IF I$<>CHR$(13)AND I$<>CHR$(20
     )THEN 2120                                   <136>
2130 IF I$=CHR$(20)THEN 2080                      <057>
2140 C=1:GOSUB 10:IF Y=0 THEN 2010                <132>
3000 REM * RECHNUNG *                             <043>
3010 IF F=1 THEN A(12)=A(9)*(1/(1+(A(10)/1
     00))↑A(11))                                  <039>
3020 IF F=2 THEN A(12)=A(9)/(1/(1+(A(10)/1
     00))↑A(11))                                  <083>
3030 IF F=3 THEN A(12)=((A(9)/A(10))↑(1/A(
     11))-1)*100                                  <246>
3040 IF F=4 THEN A(12)=LOG(A(9)/A(10))/LOG
     (1+(A(11)/100))                              <234>
3500 REM * BILDSCHIRM                             <233>
3510 Z=3:S=13:GOSUB 90:PRINT"EINGABE :":Z=
     4:GOSUB 90:PRINT"TTTTTTTTTT"                 <122>
3520 Z=5:FOR I=9 TO 11:GOSUB 90:PRINT A$(I
     )" :"A(I):Z=Z+2:NEXT                         <016>
3530 Z=13:GOSUB 90:PRINT"RESULTAT :":Z=14:
     GOSUB 90:PRINT"TTTTTTTTTT":GOSUB 70          <064>
3540 Z=16:GOSUB 90:PRINT"{RVSON}"A$(F)"{RV
     OFF,SPACE}:";:PRINT TAB(38-LEN(AB$));
     AB$                                          <149>
3550 PRINT TAB(39-LEN(AB$));:FOR I=1 TO LE
     N(AB$)-1:PRINT"=";:NEXT                      <170>
3560 Z=20:GOSUB 90:IF K=1 THEN PRINT"AUSDR
     UCK IN DM{SHIFT-SPACE}!":GOTO 3580           <244>
3570 IF K=0 THEN PRINT"AUSDRUCK IN PROZENT
     !"                                           <255>
3580 Z=23:S=0:GOSUB 90:FOR I=1 TO 4:PRINT"
     {RVSON}"BA$(I)"{RVOFF}";:NEXT                <234>
3590 Y=1:GOSUB 200:IF Y=4 THEN 80                 <041>
3600 IF Y=3 THEN 1010                             <136>
3610 IF Y=2 THEN 4500                             <247>
4000 REM * AENDERUNG *                            <039>
4010 C=1:GOSUB 10                                 <239>
4020 Z=3:S=13:GOSUB 90:PRINT"AENDERUNG :":
     Z=4:GOSUB 90:PRINT"TTTTTTTTTTT"              <139>
4030 Z=5:S=13:FOR I=9 TO 11:GOSUB 90:PRINT
     "{RVSON}"A$(I)"{RVOFF,SPACE}:";A(I):Z
     =Z+2:NEXT                                    <030>
4040 Z=23:S=0:GOSUB 90:PRINT MD$;                 <059>
4050 Z=5:Z1=Z:Z2=9:S=13:Y=9:Y1=Y:Y2=11:SW=
     2:C=1:GOSUB 100                              <253>
4060 IF C=0 THEN GOSUB 10:GOTO 3000               <068>
4070 Z=13:S=13:GOSUB 90:PRINT"{RVSON}"A$(Y
     )"{RVOFF,SPACE}: "PM$:GOSUB 95               <010>
4080 Z=23:S=0:GOSUB 90:PRINT MC$;:Z=13:S=2
     7:GOSUB 300:A(Y)=VAL(IN$):GOTO 4010          <211>
4500 REM * AUSDRUCK *                             <254>
4505 TA$=CHR$(10):TB$=CHR$(16)                    <198>
4510 OPEN 4,4,7:PRINT#4,CHR$(14);TB$"10DIS
     KONTIEREN EINES{SHIFT-SPACE}KAPITALS"        <246>
4520 PRINT#4,TB$"10-----------------------
     ---"TA$                                      <014>
4530 FOR I=9 TO 11:PRINT#4,CHR$(15);TB$"10
     *** "A$(I)" : ";A(I):NEXT                    <116>
4540 GOSUB 70:PRINT#4,TA$;TB$"14"A$(F)" :
     "AB$                                         <037>
4550 PRINT#4,TB$"15";:FOR I=1 TO 2+LEN(A$(
     F)+AB$):PRINT#4,"=";:NEXT                    <076>
4560 PRINT#4,TA$;TA$;TA$                          <075>
4570 PRINT#4,CHR$(13);TB$"10***{3SPACE}END
     KAPITAL UND BARWERT WERDEN";                 <024>
4580 PRINT#4," IN ' DM ' AUSGEDRUCKT,"            <135>
4590 PRINT#4,TB$"16DER ZINSSATZ IN ' % ' !
     "                                            <003>
4600 PRINT#4:FOR I=1 TO 40:PRINT#4,"==";:N
     EXT:PRINT#4,TA$;TA$;TA$;TA$                  <169>
4610 CLOSE 4:FOR I=56176 TO 56215:POKE I,0
     :NEXT:GOTO 3580                              <180>
5000 REM * VARIABLENLISTE *                       <016>
5010 MA$="{CLR,RVSON,6SPACE}DISKONTIEREN{S
     HIFT-SPACE}EINES{SHIFT-SPACE}KAPITALS
     {6SPACE,RVOFF}"                              <130>
5020 MB$="{RVSON,SPACE}F3 U. F5 - WAHL ! '
     RETURN' - EINGABE !{RVOFF}"                  <247>
5030 MC$="{RVSON,2SPACE}F1 - NEU !{6SPACE}
     GEBEN SIE EIN !{6SPACE,RVOFF}"               <135>
5040 MD$="{RVSON,SPACE}F1 - RESULTAT !{5SP
     ACE}F3 U. F5 - WAHL !{SPACE,RVOFF}"          <167>
5050 KN$=" ENDKAPITAL":P$="{2SPACE}ZINSSAT
     Z ":JA$="{2SPACE}JAHRE{4SPACE}":PM$="
     .........."                                  <031>
5060 KB$="{2SPACE}BARWERT{2SPACE}"                <077>
5110 BA$(1)="{2SPACE}AENDERN "                    <245>
5120 BA$(2)="{2SPACE}KOPIE{3SPACE}"               <209>
5130 BA$(3)="{2SPACE}NEU{5SPACE}"                 <224>
5140 BA$(4)=" MENUE{4SPACE}"                      <126>
5510 A$(1)="{2SPACE}BARWERT{2SPACE}"              <092>
5520 A$(2)=" ENDKAPITAL"                          <255>
5530 A$(3)="{2SPACE}ZINSSATZ "                    <154>
5540 A$(4)="{2SPACE}LAUFZEIT "                    <069>
5550 A$(5)="{2SPACE}MENUE{4SPACE}"                <146>
5590 A$(13)="{2SPACE}AENDERN{2SPACE}"             <205>
5600 A$(14)="{2SPACE}RESULTAT "                   <113>
5610 A$(15)="{2SPACE}MENUE{4SPACE}":RETURN        <097>
```


Listing 4. Diskontieren eines Kapitals (Schluß)

```
7 POKE 45,PEEK(174):POKE 46,PEEK(175):CLR  <013>
8 REM BY KLAUS KLOEKER                     <245>
9 CLR:DIM A$(15),E(50,6),E$(50,6):GOTO 100
  0                                        <225>
10 Z1=3:Z2=9:S=13:REM * MASKE LOESCHEN *   <217>
15 FOR Z=Z1 TO Z2:GOSUB 90:PRINT"{26SPACE}
   ":NEXT:IF C=0 THEN RETURN               <088>
20 Z1=13:Z2=21:C=0:GOTO 15                 <130>
30 FOR J=2 TO 6:E$(I,J)=STR$(INT(E(I,J)*10
   +.5)):REM * FORMAT TABELLE              <158>
35 E$(I,J)=LEFT$(E$(I,J),LEN(E$(I,J))-1)+"
   ."+RIGHT$(E$(I,J),1)                    <227>
40 L$="":FOR A=1 TO 10-LEN(E$(I,J)):L$=L$+
   ".":NEXT                                <133>
45 E$(I,J)=L$+E$(I,J)                      <160>
50 IF LEN(E$(I,J))>10 THEN E$(I,J)=RIGHT$(
   E$(I,J),10)                             <000>
55 NEXT:RETURN                             <250>
70 LOAD"MENUE",8:REM *LOAD MENUE           <252>
80 A=55909:FOR I=1637 TO 1662:POKE I,100:P
   OKE A,1:A=A+1:NEXT:RETURN               <025>
90 POKE 214,Z:POKE 211,S:SYS 58732:RETURN:
   REM * CURSORPOSITION *                  <003>
95 FOR I=56216 TO 56255:POKE I,0:NEXT:RETU
   RN                                      <003>
100 REM * AUSWAHLROUTINE *                 <155>
110 GOSUB 90:PRINT"{RVSON,11SPACE,RVOFF}":
    FOR I=1 TO 75:NEXT I                   <075>
120 GOSUB 90:PRINT"{RVSON}"A$(Y)"{RVOFF}":
    FOR I=1 TO 125:NEXT                    <112>
130 GET G$:IF G$<>CHR$(133)AND G$<>CHR$(13
    4)AND G$<>CHR$(135)AND G$<>CHR$(13)THE
    N 110                                  <023>
140 IF G$=CHR$(133)AND C THEN C=0:RETURN   <151>
150 IF G$=CHR$(13)THEN GOSUB 90:A$(Y)=">"+
    MID$(A$(Y),2,10):PRINT"{RVSON}"A$(Y)"{
    RVOFF}":RETURN                         <216>
155 IF G$=CHR$(134)THEN 170                <213>
160 IF G$=CHR$(135)THEN 190                <095>
165 GOTO 110                               <109>
170 Z=Z+SW:Y=Y+1:IF Z>Z2 THEN Z=Z1:Y=Y1    <115>
180 GOTO 110                               <124>
190 Z=Z-SW:Y=Y-1:IF Z<Z1 THEN Z=Z2:Y=Y2    <212>
195 GOTO 110                               <139>
200 REM * WAHL-FUSSZEILE *:S=0             <128>
210 Z=23:GOSUB 90:PRINT"{RVSON,8SPACE,RVOF
    F}";:FOR I=1 TO 75:NEXT                <218>
220 GOSUB 90:PRINT"{RVSON}"BA$(Y)"{RVOFF}"
    ;:FOR I=1 TO 175:NEXT                  <141>
230 GET G$:IF G$<>CHR$(134)AND G$<>CHR$(13
    5)AND G$<>CHR$(13)THEN 210             <217>
240 IF G$=CHR$(13)THEN GOSUB 90:PRINT"{RVS
    ON}{RVOFF}";:RETURN                    <145>
250 IF G$=CHR$(135)THEN 280                <153>
260 S=S+8:Y=Y+1:IF S>32 THEN S=0:Y=1       <076>
270 GOTO 210                               <224>
280 S=S-8:Y=Y-1:IF S<0 THEN S=32:Y=5       <077>
290 GOTO 210                               <244>
300 REM * EINGABEROUTINE *                 <179>
310 GOSUB 90                               <058>
320 IN$=""                                 <010>
330 PRINT"T{LEFT}";                        <197>
340 GET G$:IF G$=""THEN 340                <075>
350 G=ASC(G$):IF G=13 THEN GOTO 460        <165>
360 IF G=13 THEN RETURN                    <080>
370 IF G$<"0"OR G$>"9"THEN 420             <011>
380 IN$=IN$+G$                             <221>
390 PRINT G$;                              <209>
400 IF LEN(IN$)>N THEN FOR I=1 TO LEN(IN$)
    :PRINT CHR$(20);:NEXT:GOTO 320         <227>
410 GOTO 330                               <148>
420 IF G=46 THEN 380                       <130>
425 IF G$=CHR$(133)THEN 9                  <253>
430 IF G<>20 THEN 330                      <135>
440 IF LEN(IN$)<1 THEN 330                 <140>
450 IN$=LEFT$(IN$,LEN(IN$)-1):GOTO 390     <227>
460 PN$="":FOR I=1 TO(9-LEN(IN$)):PN$=PN$+
    ".":NEXT:PRINT PN$                     <230>
470 IF LEN(IN$)<1 THEN 310                 <106>
480 RETURN                                 <028>
900 REM * MASKE *                          <039>
910 PRINT MA$:A=55388:FOR I=1076 TO 1906 S
    TEP 40:POKE I,101:POKE A,1:A=A+40:NEXT <226>
920 A=55776:FOR I=1504 TO 1514:POKE I,100:
    POKE A,1:A=A+1:NEXT                    <046>
930 A=55785:FOR I=1517 TO 1546:POKE I,100:
    POKE A,1:A=A+1:NEXT                    <171>
950 Z=3:S=0:GOSUB 90:PRINT"AUSWAHL :":PRIN
    T"TTTTTTTTT":PRINT"{RVSON,SPACE}- - -
    - -{SPACE,RVOFF}"                      <245>
960 Z=6:FOR I=1 TO 3:GOSUB 90:PRINT"{RVSON
    }"A$(I)"{RVOFF}":PRINT"{RVSON,SPACE}-
    - - - -{SPACE,RVOFF}":Z=Z+2:NEXT       <236>
970 RETURN                                 <010>
999 :                                      <213>
1000 REM ** TILGUNGSRECHNUNG *             <090>
1010 GOSUB 6000:GOSUB 900                  <081>
1020 Z=23:S=0:GOSUB 90:PRINT MB$           <122>
1080 Z=6:Z1=Z:Z2=10:S=0:Y=1:Y1=Y:Y2=3:SW=2
     :C=0:GOSUB 100:F=Y:IF F=3 THEN 70     <128>
2000 REM * EINGABEMASKE *                  <128>
2010 Z=3:S=13:GOSUB 90:PRINT"EINGABE :":Z=
     4:GOSUB 90:PRINT"TTTTTTTTT"           <033>
2020 Z=5:FOR I=4 TO 6:S=13:GOSUB 90:PRINT"
     {RVSON}"A$(I)"{RVOFF,SPACE}: ";PM$    <067>
2030 Z=Z+2:NEXT:Z=23:S=0:GOSUB 90:PRINT MC
     $                                     <013>
2040 Z=5:S=27:N=9:FOR J=4 TO 6:GOSUB 300:A
     (J)=VAL(IN$):Z=Z+2:NEXT               <001>
2070 Z=14:S=13:GOSUB 90:PRINT"EINGABE KORR
     EKT ? (J/N):":GOSUB 80                <074>
2080 Z=14:S=38:GOSUB 90:PRINT"T{LEFT}";    <081>
2090 GET G$:IF G$<>"J"AND G$<>"N"THEN 2090 <206>
2100 PRINT G$:IF G$="J"THEN Y=1            <075>
2110 IF G$="N"THEN Y=0                     <107>
2120 GET I$:IF I$<>CHR$(13)AND I$<>CHR$(20
     )THEN 2120                            <136>
2130 IF I$=CHR$(20)THEN 2080               <057>
2140 C=1:GOSUB 10:IF Y=0 THEN 2010         <132>
3000 REM * RECHNUNG *                      <043>
3005 PRINT"{CLR}":Z=23:S=23:GOSUB 90:PRINT
     MG$:EE(3)=0:EE(5)=0                   <171>
3010 IF F=2 THEN 3500                      <128>
3100 REM * RATENTILGUNG *                  <194>
3110 T1=A(4)/A(6):J2=0:T2=0                <011>
3120 FOR I=1 TO A(6):R1=A(4)-J2*T2:Z1=R1*(
     A(5)/100):A1=Z1+T1                    <144>
3130 E(I,2)=R1:E(I,3)=Z1:E(I,4)=T1:E(I,5)=
     A1:E(I,6)=A1/12:GOSUB 30              <254>
3140 FOR J=3 TO 5 STEP 2:EE(J)=EE(J)+E(I,J
     ):NEXT                                <130>
3150 J2=J2+1:T2=(A(4)/A(6)):NEXT:GOTO 3700 <172>
3500 REM * AUNNUITAETENTILGUNG *           <186>
3510 Q1=A(5)/100+1:T1=A(4)*((Q1-1)/(Q1↑A(6
     )-1))                                 <206>
3520 Z1=A(4)*(A(5)/100):A1=T1+Z1:R1=A(4)   <128>
3530 E(1,2)=R1:E(1,3)=Z1:E(1,4)=T1:E(1,5)=
     A1:E(1,6)=A1/12:I=1:GOSUB 30          <251>
3540 T2=T1:FOR I=2 TO A(6):R1=R1-T2:Z2=R1*
     (A(5)/100):T2=T2*Q1                   <139>
3550 E(I,2)=R1:E(I,3)=Z2:E(I,4)=T2:E(I,5)=
     A1:E(I,6)=A1/12:GOSUB 30              <244>
3560 FOR J=3 TO 5 STEP 2:EE(J)=EE(J)+E(I,J
     ):NEXT J:NEXT I                       <177>
3570 EE(3)=EE(3)+E(1,3):EE(5)=EE(5)+E(1,5) <060>
3700 EE(3)=EE(3)/2:EE(5)=EE(5)/2           <011>
3710 FOR I=3 TO 5 STEP 2:EE$(I)=STR$(INT(E
     E(I)*10/.5))                          <161>
3720 EE$(I)=LEFT$(EE$(I),LEN(EE$(I))-1)+".
     "+RIGHT$(EE$(I),1):NEXT               <217>
3800 REM * SCREEN *                        <223>
3805 GOSUB 900                             <001>
3810 Z=3:S=13:GOSUB 90:PRINT"EINGABE :":Z=
     4:GOSUB 90:PRINT"TTTTTTTTT"           <055>
3820 Z=5:S=13:FOR I=4 TO 6:GOSUB 90:PRINT
     A$(I)" :"A(I):Z=Z+2:NEXT              <210>
3830 Z=14:S=13:GOSUB 90:PRINT"RESULTAT :":
```

Listing 5. Erstellung eines Tilgungsplans

```
     Z=15:GOSUB 90:PRINT"TTTTTTTTTTT"          <018>
3840 Z=17:FOR J=3 TO 5 STEP 2:GOSUB 90:PRI
     NT"{RVSON}"EC$(J)"{RVOFF}:"EE$(J):Z=Z
     +2:NEXT                                    <223>
3850 Z=21:GOSUB 90:PRINT"AUSDRUCK IN 'DM
     !"                                         <232>
3860 ZA=VAL(EE$(3))/(VAL(EE$(5))/100)           <157>
3870 Z=14:S=0:GOSUB 90:PRINT"{RVSON,3SPACE
     }ZINS-{3SPACE,RVOFF}":Z=15:GOSUB 90:P
     RINT"{RVSON,3SPACE}ANTEIL{2SPACE,RVOF
     F}"                                        <219>
3880 ZA$=STR$(ZA):IF ZA<10 THEN ZA$=" "+ZA
     $                                          <149>
3890 ZA$=LEFT$(ZA$,5):Z=17:S=2:GOSUB 90:PR
     INT ZA$;" %"                               <144>
3900 Z=18:S=3:GOSUB 90:PRINT"======"            <091>
3910 Z=23:S=0:GOSUB 90:FOR I=1 TO 5:PRINT"
     {RVSON}"BA$(I)"{RVOFF}";:NEXT              <072>
3920 Y=1:GOSUB 200:IF Y=5 THEN 70               <180>
3930 IF Y=4 THEN 9                              <166>
3940 IF Y=3 THEN 5500                           <227>
3950 IF Y=2 THEN 5000                           <200>
4000 REM * TABELLE *                            <054>
4010 GOSUB 95:X1=2:X2=4:Y1=1:Y2=A(6)            <229>
4020 Z=1:S=0:GOSUB 90:FOR I=1 TO 22:PRINT
     MD$:NEXT:Z=22:GOSUB 90:PRINT MH$           <187>
4030 Z=3:S=0:GOSUB 90:PRINT M$(1)               <251>
4040 Z=3:S=6:FOR I=X1 TO X2:GOSUB 90:PRINT
      M$(I):S=S+11:NEXT                         <201>
4050 Z=4:S=1:GOSUB 90:FOR I=1 TO 37:PRINT"
     =";:NEXT                                   <109>
4100 Z=6:S=1:FOR I=Y1 TO Y2                     <159>
4105 I$=STR$(I):IF I<10 THEN I$="."+I$          <057>
4107 GOSUB 90:PRINT"."+I$:S=6                   <061>
4110 FOR J=X1 TO X2                             <022>
4120 GOSUB 90:PRINT E$(I,J)                     <076>
4130 S=S+11:NEXT J                              <250>
4140 IF PEEK(214)>15 THEN Y2=I:Y1=Y2-9:GOT
     O 4200                                     <076>
4150 Z=Z+1:S=1:NEXT I                           <085>
4200 Z=17:S=1:GOSUB 90:FOR J=1 TO 38:PRINT
     "-";:NEXT                                  <001>
4500 WAIT 203,63:GET G$                         <201>
4510 IF G$=CHR$(29)THEN 4550                    <152>
4520 IF G$=CHR$(157)THEN 4580                   <040>
4530 IF G$=CHR$(17)THEN 4610                    <198>
4540 IF G$=CHR$(145)THEN 4640                   <112>
4543 IF G$=CHR$(133)THEN 3800                   <200>
4545 GOTO 4500                                  <015>
4550 X1=X1+1:X2=X2+1:REM * ->                   <036>
4560 IF X2>6 THEN X2=6:X1=4:GOTO 4500           <249>
4570 GOTO 4030                                  <056>
4580 X1=X1-1:X2=X2-1:REM * <-                   <017>
4590 IF X1<2 THEN X1=2:X2=4:GOTO 4500           <100>
4600 GOTO 4030                                  <086>
4610 IF A(6)<10 THEN GOTO 4500                  <081>
4615 Y1=Y1+10:Y2=Y2+10:REM * DOWN               <192>
4620 IF Y2>A(6)THEN Y2=A(6):Y1=Y2-9:IF I=A
     (6)THEN 4500                               <159>
4630 GOTO 4030                                  <118>
4640 IF A(6)<10 THEN GOTO 4500                  <038>
4645 Y1=Y1-10:Y2=Y2-10:REM * UP                 <049>
4650 IF Y1<1 THEN Y1=1:Y2=Y1+9:IF I<11 THE
     N 4500                                     <083>
4660 GOTO 4030                                  <148>
5000 REM * AENDERUNG *                          <023>
5010 C=1:GOSUB 10:S=0:FOR Z=13 TO 20:GOSUB
      90:PRINT"{11SPACE}":NEXT                  <016>
5020 Z=3:S=13:GOSUB 90:PRINT"AENDERUNG :":
     Z=4:GOSUB 90:PRINT"TTTTTTTTTTT"           <123>
5030 Z=5:S=13:FOR I=4 TO 6:GOSUB 90:PRINT"
     {RVSON}"A$(I)"{RVOFF,SPACE}:";A(I):Z=
     Z+2:NEXT                                   <000>
5040 Z=23:S=0:GOSUB 90:PRINT MF$;               <047>
5050 Z=5:Z1=Z:Z2=9:S=13:Y=4:Y1=Y:Y2=6:SW=2
     :C=1:GOSUB 100                             <086>
5060 IF C=0 GOTO 3000                           <174>
5070 Z=14:S=13:GOSUB 90:PRINT"{RVSON}"A$(Y
     )"{RVOFF,SPACE}: ";PM$:GOSUB 80            <124>
5080 Z=23:S=0:GOSUB 90:PRINT MC$;:Z=14:S=2
     7:GOSUB 300:A(Y)=VAL(IN$)                  <106>
5090 A$(Y)=" "+MID$(A$(Y),2,10):GOTO 5010       <218>
5500 REM * PRINTER                              <036>
5505 TB$=CHR$(16):TA$=CHR$(10)                  <122>
5510 OPEN 4,4,7:PRINT#4,CHR$(14)TB$"10AUSD
     RUCK EINES TILGUNGSPLANES"                 <030>
5520 PRINT#4,TB$;"10-----------------------
     -------"TA$                                <201>
5530 FOR I=4 TO 6:PRINT#4,CHR$(15)TB$"10**
     * "A$(I)" : "A(I):NEXT                     <105>
5540 PRINT#4,TA$;TA$;TA$                        <039>
5550 FOR I=1 TO 6:PRINT#4,"{3SPACE}"M$(I);
     :NEXT:PRINT#4," "                          <077>
5560 FOR I=1 TO 40:PRINT#4,"--";:NEXT:PRIN
     T#4                                        <171>
5570 FOR I=1 TO A(6):I$=STR$(I):IF I<10 TH
     EN I$="."+I$                               <181>
5575 PRINT#4,"{4SPACE}."+I$" ";                 <252>
5580 FOR J=2 TO 6:PRINT#4," .."E$(I,J);:NE
     XT J:PRINT#4," ":NEXT I                    <217>
5590 FOR I=1 TO 40:PRINT#4,"--";:NEXT:PRIN
     T#4                                        <201>
5595 PRINT#4," GESAMT:";                        <002>
5600 PO=27:FOR J=3 TO 5 STEP 2:PRINT#4,SPC
     (PO-LEN(EE$(J)))EE$(J);:PO=26:NEXT         <176>
5610 PRINT#4,TA$;TA$;TB$"10*** AUSDRUCK DE
     R TABELLE ERFOLGT IN 'DM' !"               <033>
5620 PRINT#4,TB$"10*** BERECHNUNG DER ZINS
     EN ERFOLGT AM ENDE DES JAHRES !"           <220>
5630 PRINT#4,TA$;TB$"10*** TILGUNG KONSTAN
     T{3SPACE}= RATENTILGUNG !!"                <143>
5640 PRINT#4,TB$"10*** ANNUITAET KONSTANT
     = ANNUITAETENTILGUNG !!"                   <003>
5650 PRINT#4:FOR I=1 TO 40:PRINT#4,"==";:N
     EXT I:PRINT#4,TA$;TA$;TA$;TA$              <169>
5660 CLOSE 4:GOTO 3860                          <189>
6000 REM * VARIABLEN *                          <059>
6010 MA$="{CLR,RVSON,4SPACE}T I{SHIFT-SPAC
     E}L{SHIFT-SPACE}G{SHIFT-SPACE}U{SHIFT
     -SPACE}N{SHIFT-SPACE}G{SHIFT-SPACE}S{
     SHIFT-SPACE}R{SHIFT-SPACE}E{SHIFT-SPA
     CE}C{SHIFT-SPACE}H{SHIFT-SPACE}N{SHIF
     T-SPACE}U{SHIFT-SPACE}N{SHIFT-SPACE}G
     {4SPACE,RVOFF}"                            <202>
6020 MB$="{RVSON,SPACE}F3 U. F5 - WAHL ! '
     RETURN' - EINGABE !"                       <081>
6030 MC$="{RVSON,SPACE}F1 - AUSWAHL !{3SPA
     CE}GEBEN SIE EIN !{6SPACE,RVOFF}"          <188>
6040 MD$="{39SPACE}"                            <078>
6050 ME$="{RVSON,39SPACE,RVOFF}"                <046>
6060 MF$="{RVSON,SPACE}F1 - RESULTAT !{5SP
     ACE}F3 U. F5 - WAHL !{SPACE,RVOFF}"        <187>
6070 MG$="BITTE WARTEN !!!"                     <103>
6080 MH$="{RVSON,SPACE}F1 - RESULTAT !{4SP
     ACE}CURSOR - STEUERN !{SPACE,RVOFF}"       <221>
6100 A$(1)=" RATENTILG."                        <090>
6110 A$(2)=" ANNU.TILG."                        <099>
6120 A$(3)="{2SPACE}MENUE{4SPACE}"              <144>
6130 A$(4)=" DARLEHEN{2SPACE}"                  <229>
6140 A$(5)=" ZINSSATZ{2SPACE}"                  <064>
6150 A$(6)=" JAHRE{5SPACE}"                     <154>
6160 PM$="........."                            <064>
6200 M$(1)=" JAHR "                             <112>
6210 M$(2)="RESTSCHULD"                         <091>
6220 M$(3)="{4SPACE}ZINSEN"                     <076>
6230 M$(4)="{3SPACE}TILGUNG"                    <079>
6240 M$(5)=" ANNUITAET"                         <169>
6250 M$(6)=" ANNU./MON"                         <105>
6300 EC$(3)=" ZINSEN{2SPACE}GES."               <196>
6310 EC$(5)=" ANNUIT. GES."                     <055>
6400 BA$(1)=" TABELLE"                          <077>
6410 BA$(2)=" AENDERN"                          <085>
6420 BA$(3)="{2SPACE}KOPIE "                    <047>
6430 BA$(4)="{3SPACE}NEU{2SPACE}"               <063>
6440 BA$(5)="{2SPACE}MENUE "                    <220>
6500 RETURN                                     <208>
```

© 64'er

Listing 5. Erstellung eines Tilgungsplans (Schluß)

Schachmeister

Viele Schachspieler zeichnen ihre Partien auf und überprüfen sie hinterher. Der C 64 hilft dabei. Die Verbesserung aus Ausgabe 4/85 ist in die Anleitung und das Listing integriert.

Nach dem Start des Programms muß man einen Augenblick warten, bis der Computer die neuen Multicolor-Zeichen definiert hat und diverse Felder und Listen aufgestellt hat (zirka 20 Sekunden). Daraufhin kann man eine der gespeicherten Partien auswählen und das Schachbrett baut sich auf.

Am rechten Rand werden die Namen der Spieler, das Jahr, die Zugnummer und der Zug sowie verschiedene Meldungen wie »Schach«, »Matt«, »Rochade« ... ausgegeben. Das Schachbrett nimmt zirka ¾ des Bildschirms ein.

Durch Drücken der F7-Taste nach einem Zug wird das Bild bis zu einem erneuten Tastendruck »eingefroren«. Durch Drücken einer anderen Taste nach einem Zug wird der Ablauf beschleunigt. Durch anhaltendes Drücken der Space-Taste kann man so eine Art »Blitzschach« erreichen. Nach Spielende führt ein Tastendruck wieder zum Auswahlmenü.

Eingeben eigener Schachpartien:

1. Zuerst werden die Namen der Spieler und das Jahr eingegeben:
10000 DATA SPIELER 1, SPIELER 2, 1984.
2.Die Eingabe der Züge:
Beispiel für einen normalen Zug: 10010 DATA E2E4, E7E5, G1F3 ...
Wenn eine Figur eine gegnerische schlägt, braucht man das nicht anzugeben; der Computer erkennt das von selbst.
3. Sonderregeln (Beispiele):
Aufgabe eines Spielers:
10020 DATA ...,SA
(Schwarz gibt auf)
10020 DATA ..., WA
(Weiß gibt auf)
Rochaden:
10020 DATA ..., KR
(kurze Rochade)
10020 DATA ..., LR
(lange Rochade)
Es braucht nicht angegeben zu werden, welcher der beiden Spieler rochiert.
Schach, Matt, Remis:
Um dem Computer einen dieser drei Zustände anzuzeigen, muß ein S (Schach), ein M (Matt) oder ein R (Remis) dem jeweiligen Zug angehängt werden:
10020 DATA ..., D6E6S
(Schach wird geboten)
10030 DATA ..., F6E6M
(Schachmatt)
10020 DATA ..., E5E6R
(Remis durch Übereinkunft der Spieler)
10020 DATA B6B7P (Remis durch Patt)
10020 DATA C4C5W (Remis durch dreimal wiederholte Stellung)
10020 DATA A1C1Z (Remis durch 50-Züge-Regel. Es wurde seit 50 Zügen kein Bauernzug gemacht)
Erreicht ein Bauer die gegnerische Grundlinie, so wird er in eine andere Figur umgewandelt. Und zwar:
in einen Turm:
DATA A7A8T
in einen Springer (engl. Horse)
DATA A7A8H
in einen Läufer:
DATA A7A8C
Wenn ein Bauer in eine Dame umgetauscht wird, so ist dem jeweiligen Zug ein D anzufügen:
10020 DATA ..., A7A8D
(weißer Bauer wird in Dame getauscht)
Wird ein gegnerischer Bauer en Passant geschlagen, so ist dem entsprechenden Zug ein »v« anzuhängen. (enpassant = im Vorübergehen)

Wichtig:
Nach Ende jeder Partie muß unbedingt ein E angefügt werden, damit der Computer eine Liste der Partien aufstellt. Nach der letzten Partie ist statt dem E ein X anzufügen:
19999 DATA ...,F6E6M,E
19999 DATA ...,F6E6M,X

Die Programmlänge ohne gespeicherte Partien beträgt zirka 7,5 KByte; pro gespeicherter Partie werden zusätzlich zirka 0,5 KByte benötigt.

Da der Bildschirmspeicher verschoben wird, muß nach einem RUN STOP-RESTORE blind RUN eingetippt werden, um das Programm wieder auflisten zu lassen.

(Thomas Behrend/tr)

A$-Feld	: Speicherung der sechs Schachfiguren
A(x,y)-Feld	: Spielfeld (Figuren)
B(x,y)-Feld	: Spielbrett
P-Feld	: Zähler zum Überspringen der DATA-Zeilen bis zur gewählten Partie
P$-Feld	: Namen der Spieler (weiß)
P1$-Feld	: Namen der Spieler (schwarz)
P	: Anzahl der gespeicherten Partien
DA	: Zähler zum Überspringen der ersten DATA-Zeilen
AW	: Flag für verschiedene Spielzustände (Matt, Remis...)

Wichtige Programmzeilen und Programmteile:

150- 190	: MC-Programm zur Beschleunigung der Zeichendefinition
200- 260	: Zeichendefinition
300- 860	: Zeichen-DATA (Multicolor)
900-1050	: Felder aufstellen und Variablen definieren
1100,1110	: Multicolor-Modus einschalten
1170-1250	: Partienanfänge einlesen und Tabelle aufstellen
1290-1350	: Auswahlmenü
1390-1660	: Bildschirmaufbau
1730-1910	: Hauptprogramm (normaler Zug)
1950-2410	: Besondere Regeln (Rochade, Dametausch, Matt, Remis...)
2470-2490	: Einzelnes Feld löschen
2530-2580	: Figur drucken
2620	: Koordinaten errechnen
2660-2720	: Zeitschlaufe mit Unterbrechung und Verkürzung
2760-2800	: Grundstellung der Figuren
2840-2850	: Bildschirmpositionierung
2890-2910	: Spielende
10000-	: DATAs für die Partien

Alle REM-Zeilen dienen nur zur Anschaulichkeit und können entfernt werden.

Tabelle: Programmbeschreibung zum »Schachmeister«

```
10 REM ********************************     <060>
20 REM *       >> SCHACHMEISTER <<      *   <098>
30 REM *                                *   <079>
40 REM * THOMAS BEHREND                 *   <118>
50 REM * IN DER WANNE 16     7900 ULM   *   <107>
60 REM *       (8.1984)                 *   <219>
70 REM ********************************     <120>
80 REM                                      <142>
90 REM  -- ALLE REM-ZEILEN KOENNEN  --      <107>
100 REM -- WEGGELASSEN WERDEN       --      <040>
110 REM                                     <172>
120 REM -- NEUE ZEICHEN --                  <073>
130 REM                                     <192>
140 POKE 53281,6:POKE 53280,1:PRINT"{CLR}"
    CHR$(5)                                 <114>
150 DATA 120,169,51,133,1,169,0,133,95,133
    ,90,133,88,169,208,133,96,169,240,133   <105>
160 DATA 89,169,224,133,91,32,191,163,169,
    55,133,1,88,96                          <172>
170 REM -- MC-PROGRAMM --                   <106>
180 J=4097:FOR I=832 TO 865:READ A:J=J-A:P
    OKE I,A:NEXT                            <184>
190 IF J<>0 THEN PRINT"DATA-FEHLER IN ZEIL
    E 15-20":STOP                           <206>
200 SYS 832:POKE 850,160:SYS 832:POKE 5657
    6,PEEK(56576)AND 252:POKE 53272,8       <154>
210 POKE 648,192:PRINT"{DOWN}"TAB(5)"> SCH
    ACHMEISTER <"                           <068>
220 PRINT"{2DOWN}"TAB(5)"BITTE EINEN MOMEN
    T WARTEN..."                            <011>
230 AD=57344:FOR I=64 TO 91:B=AD+I*8:FOR J
    =0 TO 7:READ C:POKE B+J,C:NEXT:NEXT     <163>
240 FOR I=192 TO 219:B=AD+I*8:FOR J=0 TO 7
    :READ C:POKE B+J,C:NEXT:NEXT            <098>
250 I=175:B=AD+I*8:FOR J=0 TO 7:READ C:POK
    E B+J,C:NEXT                            <014>
260 POKE 56,130                             <010>
270 REM                                     <078>
280 REM -- DATA FUER SCHACHFIGUREN --       <118>
290 REM                                     <098>
300 DATA 0,0,60,60,255,255,60,60            <200>
310 DATA 0,0,0,0,0,0,0,0                    <230>
320 DATA 0,0,0,0,60,255,255,255             <136>
330 DATA 255,255,60,60,60,60,60,60          <094>
340 DATA 0,0,3,3,15,10,0,0                  <021>
350 DATA 60,255,255,255,255,170,0,0         <050>
360 DATA 0,0,192,192,240,160,0,0            <028>
370 DATA 0,0,3,3,3,3,3,3                    <168>
380 DATA 0,0,60,60,60,255,255,255           <243>
390 DATA 0,0,192,192,192,192,192,192        <213>
400 DATA 3,0,0,0,0,0,0,0                    <088>
410 DATA 255,60,60,60,60,60,60,60           <097>
420 DATA 255,255,255,255,255,255,255,255    <248>
430 DATA 192,0,0,0,0,0,0,0                  <168>
440 DATA 0,0,60,60,240,195,255,255          <092>
450 DATA 255,255,255,255,255,170,0,0        <137>
460 DATA 0,0,0,0,192,192,0,0                <031>
470 DATA 0,0,0,0,0,3,3,15                   <061>
480 DATA 0,48,48,63,255,207,207,255         <133>
490 DATA 0,0,0,0,192,192,192,192            <150>
500 DATA 63,63,63,60,0,0,0,3                <201>
510 DATA 255,207,15,63,63,255,255,255       <105>
520 DATA 192,192,0,0,0,0,0,192              <228>
530 DATA 3,3,3,3,15,10,0,0                  <077>
540 DATA 192,240,240,240,240,160,0,0        <070>
550 DATA 0,0,12,15,3,3,3,3                  <194>
560 DATA 0,0,48,240,192,192,192,192         <250>
570 DATA 0,0,0,0,3,3,0,0                    <100>
580 DATA 85,85,125,125,255,255,125,125      <173>
590 DATA 85,85,85,85,85,85,85,85            <166>
600 DATA 85,85,85,85,125,255,255,255        <027>
610 DATA 255,255,125,125,125,125,125,125    <132>
620 DATA 85,85,87,87,95,90,85,85            <104>
630 DATA 125,255,255,255,255,170,85,85      <255>
640 DATA 85,85,213,213,245,165,85,85        <191>
650 DATA 85,85,87,87,87,87,87,87            <192>
660 DATA 85,85,125,125,125,255,255,255      <222>
670 DATA 85,85,213,213,213,213,213,213      <127>
680 DATA 87,85,85,85,85,85,85,85            <032>
690 DATA 255,125,125,125,125,125,125,125    <080>
700 DATA 255,255,255,255,255,255,255,255    <018>
710 DATA 213,85,85,85,85,85,85,85           <078>
720 DATA 85,85,125,125,245,215,255,255      <035>
730 DATA 255,255,255,255,255,170,85,85      <155>
740 DATA 85,85,85,85,213,213,85,85          <227>
750 DATA 85,85,85,85,85,87,87,95            <215>
760 DATA 85,117,117,127,255,223,223,255     <052>
770 DATA 85,85,85,85,213,213,213,213        <059>
780 DATA 127,127,127,125,85,85,85,87        <161>
790 DATA 255,223,95,127,127,255,255,255     <089>
800 DATA 213,213,85,85,85,85,85,213         <247>
810 DATA 87,87,87,87,95,90,85,85            <073>
820 DATA 213,245,245,245,245,165,85,85      <094>
830 DATA 85,85,93,95,87,87,87,87            <056>
840 DATA 85,85,117,245,213,213,213,213      <011>
850 DATA 85,85,85,85,87,87,85,85            <190>
860 DATA 126,189,215,231,231,215,189,126    <095>
870 REM                                     <170>
880 REM -- VARIABLEN UND FELDER --          <139>
890 REM                                     <190>
895 DIM A$(12)                              <126>
900 B$(1)="ABAACADEF":REM BAUER             <241>
910 B$(2)="GHIJLMDOF":REM TURM              <103>
920 B$(3)="ANAAKADEF":REM LAEUFER           <078>
930 B$(4)="QRSTUVWOX":REM PFERD             <186>
940 B$(5)="YHZACADEF":REM DAME              <065>
950 B$(6)="[@PJCMDEF":REM KOENIG            <170>
960 FOR I=1 TO 6:FOR J=0 TO 2:FOR K=1 TO 3 <000>
970 A$=MID$(B$(I),J*3+K,1):A$(I)=A$(I)+CHR
    $(ASC(A$)+128):NEXT                     <062>
980 A$(I)=A$(I)+"{DOWN,3LEFT}":NEXT:NEXT    <217>
990 DIM B(8,8)                              <233>
1000 FOR I=7 TO 1 STEP-2:FOR J=1 TO 7 STEP
     2:B(I,J)=1:NEXT:NEXT                   <194>
1010 FOR I=8 TO 2 STEP-2:FOR J=2 TO 8 STEP
     2:B(I,J)=1:NEXT:NEXT                   <241>
1020 DIM A(8,8)                             <255>
1030 GOSUB 2760                             <026>
1040 Z1$="{RVSON,CYAN,SPACE,RED,13SPACE}"   <146>
1050 :                                      <010>
1060 REM                                    <106>
1070 REM -- MULTICOLOR-MODUS --             <226>
1080 REM -- EINSCHALTEN      --             <043>
1090 REM                                    <136>
1100 POKE 53281,6:POKE 53282,7:POKE 53283,
     5                                      <073>
1110 POKE 53270,PEEK(53270)OR 16            <026>
1120 REM                                    <166>
1130 REM -- LISTE DER PARTIEN --            <020>
1140 REM -- AUFSTELLEN        --            <201>
1150 REM                                    <196>
1160 DATA-1                                 <026>
1170 RESTORE:DA=0                           <205>
1180 DA=DA+1:READ A:IF A=-1 THEN 1200       <010>
1190 GOTO 1180                              <131>
1200 DIM P(20),P$(20),P1$(20),JA$(20):P=0:
     D=0                                    <007>
1210 P=P+1:READ P$(P),P1$(P),JA$(P):D=D+3   <155>
1220 READ A$:D=D+1:IF A$="E"OR A$="X"THEN
     1240                                   <221>
1230 GOTO 1220                              <250>
1240 IF A$="X"THEN 1290                     <195>
1250 P(P)=D:GOTO 1210                       <031>
1260 REM                                    <050>
1270 REM -- AUSWAHLMENUE --                 <005>
1280 REM                                    <072>
1290 RESTORE:FOR I=1 TO DA:READ A:NEXT      <013>
1300 PRINT CHR$(5)"{CLR,DOWN,3SPACE}- AUSW
     AHL -"                                 <145>
1310 PRINT:PRINT                            <248>
1320 FOR I=1 TO P:PRINT I;TAB(5)P$(I)TAB(1
     6)P1$(I),JA$(I):NEXT                   <156>
1330 POKE 198,0:INPUT"{2DOWN}";W:IF W<1 OR
     W>P THEN 1330                          <089>
1340 PRINT"{CLR}":IF W=1 THEN 1390          <041>
1350 FOR I=1 TO P(W-1):READ A$:NEXT         <075>
1360 REM                                    <152>
1370 REM -- SPIELBRETT --                   <045>
1380 REM                                    <172>
1390 FOR I=1 TO 24:PRINT CHR$(159)TAB(0)"{
     RVSON,SPACE}"TAB(25)Z1$:NEXT           <230>
1400 PRINT CHR$(5)"{HOME}";                 <236>
1410 FOR Y=8 TO 1 STEP-1:X=1:GOSUB 2840:PR
     INT"{DOWN,2LEFT}"Y:X=8:GOSUB 2840      <001>
1420 PRINT"{DOWN,2RIGHT}"Y:NEXT             <240>
1430 FOR Y=8 TO 1 STEP-1:FOR X=1 TO 8       <115>
1440 GOSUB 2470                             <166>
1450 NEXT:NEXT                              <183>
1460 PRINT"{HOME}"                          <170>
1470 REM                                    <006>
1480 REM -- ANFANGSSTELLUNG --              <211>
1490 REM                                    <026>
1500 FOR Y=7 TO 8:FOR X=1 TO 8              <244>
```

Listing »Schachmeister«.
Bitte mit dem Checksummer V3 eintippen.

```
1510 GOSUB 2530:NEXT:NEXT                          <153>
1520 FOR Y=1 TO 2:FOR X=1 TO 8:GOSUB 2530:
     NEXT:NEXT                                     <073>
1530 PRINT CHR$(159)"{HOME}":PRINT TAB(27)
     "{RVSON,SPACE}SPIELER 1 "                     <120>
1540 PRINT TAB(27)"{RVSON,11SPACE,RVOFF}"          <182>
1550 PRINT TAB(27)"{UP,RVSON,11SPACE}"             <062>
1560 POKE 214,18:PRINT:PRINT TAB(27)CHR$(3
     0)"{RVSON,SPACE}SPIELER 2 "                   <141>
1570 PRINT TAB(27)"{RVSON,11SPACE,RVOFF}"          <212>
1580 PRINT TAB(27)"{UP,RVSON,11SPACE}"             <092>
1590 POKE 214,5:PRINT:PRINT TAB(27)CHR$(5)
     "{RVSON}ZUG:{RVOFF,2SPACE}1"                  <249>
1600 POKE 214,7:PRINT:PRINT TAB(28)CHR$(15
     8)"{RVSON,7SPACE,RVOFF}"                      <080>
1610 POKE 214,9:PRINT:PRINT TAB(27)CHR$(5)
     "{RVSON,10SPACE}"                             <167>
1620 PRINT TAB(27)"{RVSON,10SPACE}"                <157>
1630 PRINT TAB(27)"{UP,RVSON,10SPACE}"             <142>
1640 READ A$:A$=LEFT$(A$,9):POKE 214,20:PR
     INT:PRINT TAB(28)"{RVSON}"A$                  <160>
1650 READ A$:A$=LEFT$(A$,9):POKE 214,2:PRI
     NT:PRINT TAB(28)"{RVSON}"A$                   <227>
1660 READ A$:POKE 214,15:PRINT:PRINT TAB(2
     8)A$                                          <082>
1670 ZU=1:Z1=-1:AW=0                               <167>
1680 REM                                           <218>
1690 REM ************************                  <219>
1700 REM *  -- HAUPTPROGRAMM -- *                  <200>
1710 REM ************************                  <239>
1720 REM                                           <002>
1730 PRINT"{HOME}":Z1=Z1+1:IF Z1=2 THEN Z1
     =0:ZU=ZU+1:POKE 214,5:PRINT:PRINT TAB
     (32)CHR$(5)ZU                                 <174>
1740 READ A$:IF LEN(A$)<>4 THEN 1950               <024>
1750 REM                                           <032>
1760 REM -- NORMALER ZUG --                        <067>
1770 REM                                           <052>
1780 B$=LEFT$(A$,2):GOSUB 2620:X=X2:Y=Y2           <111>
1790 B2$=B$                                        <222>
1800 B$=RIGHT$(A$,2):GOSUB 2620                    <104>
1810 O$="-":IF A(X2,Y2)<>0 THEN O$="/"             <224>
1820 A$=B2$+O$+B$                                  <054>
1830 POKE 214,7:PRINT:PRINT TAB(29)CHR$(5)
     "{RVSON}"A$                                   <201>
1840 FOR I=1 TO 4:GOSUB 2530:GOSUB 2470:NE
     XT                                            <095>
1850 A1=A(X,Y):A(X,Y)=0:A(X2,Y2)=A1                <062>
1860 X=X2:Y=Y2:FOR I=1 TO 4:GOSUB 2470:GOS
     UB 2530:NEXT                                  <139>
1865 IF AW=4 THEN GOSUB 2304:AW=1                  <017>
1870 IF AW=3 THEN AW=0:RETURN                      <172>
1880 IF AW=2 THEN 2890:REM-MATT ODER REMIS
     -                                             <033>
1890 GOSUB 2660                                    <108>
1900 IF AW=1 THEN AW=0:POKE 214,11:PRINT:P
     RINT TAB(27)"{RVSON,10SPACE}"                 <176>
1910 GOTO 1730                                     <024>
1920 REM                                           <204>
1930 REM -- BESONDERE SITUATIONEN --               <113>
1940 REM                                           <224>
1950 IF A$="E"THEN 2890                            <057>
1960 IF A$="SA"THEN 2100                           <144>
1970 IF A$="WA"THEN 2120                           <220>
1980 IF A$="KR"THEN 2190                           <210>
1990 IF A$="LR"THEN 2250                           <236>
2000 PRINT CHR$(5)                                 <244>
2010 B$=RIGHT$(A$,1):A$=LEFT$(A$,4)                <166>
2020 IF B$="S"THEN AW=1:POKE 214,11:PRINT:
     PRINT TAB(27)"{RVSON}SCHACH !":GOTO 1
     780                                           <207>
2030 IF B$="M"THEN AW=2:POKE 214,11:PRINT:
     PRINT TAB(27)"{RVSON,SPACE}MATT !":GO
     TO 1780                                       <115>
2040 IF B$="R"THEN AW=2:POKE 214,11:PRINT:
     PRINT TAB(27)"{RVSON,SPACE}REMIS !":G
     OTO 1780                                      <122>
2041 IF B$="P"THEN AW=2:POKE 214,11:PRINT:
     PRINT TAB(27)"{RVSON,SPACE}PATT !":GO
     TO 1780                                       <003>
2042 IF B$="W"THEN AW=2:POKE 214,9:PRINT:P
     RINT TAB(27)"{RVSON}DRITTE MAL"               <120>
2043 IF B$="W"THEN PRINT TAB(27)"{RVSON}GL
     EICHE",TAB(67)"{RVSON}STELLUNG !":GOT
     O 1780                                        <199>
2044 IF B$="Z"THEN AW=2:POKE 214,9:PRINT:P
     RINT TAB(27)"{RVSON,SPACE}50 ZUEGE"           <015>
2045 IF B$="Z"THEN PRINT TAB(27)"{RVSON,3S
     PACE}OHNE",TAB(67)"{RVSON,2SPACE}BAUE
     R !":GOTO 1780                                <052>
2046 IF B$="V"THEN AW=4:POKE 214,11:PRINT:
     PRINT TAB(27)"{RVSON}EN-PASSANT":GOTO
      1780                                         <013>
2050 IF B$="D"OR B$="T"OR B$="H"OR B$="L"T
     HEN F$=B$:GOTO 2340                           <243>
2060 PRINT"{CLR}DATAFEHLER !!!":STOP               <071>
2070 REM                                           <100>
2080 REM -- AUFGEBEN --                            <253>
2090 REM                                           <120>
2100 A$="{RVSON}SCHWARZ"                           <056>
2110 GOTO 2130                                     <138>
2120 A$="{RVSON}WEISS"                             <252>
2130 POKE 214,9:PRINT:PRINT TAB(29)A$              <157>
2140 PRINT TAB(27)"{RVSON}GIBT AUF !"              <116>
2150 GOTO 2890                                     <227>
2160 REM                                           <190>
2170 REM -- KURZE ROCHADE --                       <147>
2180 REM                                           <210>
2190 Y=1:T=0:IF Z1=1 THEN Y=8:T=6                  <035>
2200 A(5,Y)=0:A(8,Y)=0:A(6,Y)=8-T:A(7,Y)=1
     2-T:X=5:GOSUB 2470:X=8:GOSUB 2470             <154>
2210 X=6:GOSUB 2530:X=7:GOSUB 2530:GOTO 22
     80                                            <068>
2220 REM                                           <250>
2230 REM -- LANGE ROCHADE --                       <213>
2240 REM                                           <014>
2250 Y=1:T=0:IF Z1=1 THEN Y=8:T=6                  <095>
2260 A(1,Y)=0:A(5,Y)=0:A(3,Y)=12-T:A(4,Y)=
     8-T                                           <008>
2270 X=1:GOSUB 2470:X=5:GOSUB 2470:X=3:GOS
     UB 2530:X=4:GOSUB 2530                        <071>
2280 POKE 214,11:PRINT:PRINT TAB(29)CHR$(5
     )"{RVSON}ROCHADE"                             <044>
2290 GOSUB 2660                                    <254>
2300 POKE 214,11:PRINT:PRINT TAB(29)"{RVSO
     N,7SPACE}":GOTO 1730                          <092>
2301 REM                                           <075>
2302 REM -- EN-PASSANT --                          <198>
2303 REM                                           <077>
2304 IF A1=7 THEN Y=Y-1                            <230>
2305 IF A1=1 THEN Y=Y+1                            <094>
2306 A(X,Y)=0:GOSUB 2470:RETURN                    <153>
2310 REM                                           <086>
2320 REM -- TAUSCH AUF DER GRUNDLINIE --           <025>
2330 REM                                           <106>
2340 AW=3:GOSUB 1780                               <218>
2350 IF(Y2=8)AND(F$="D")THEN A(X2,8)=11            <041>
2351 IF(Y2=1)AND(F$="D")THEN A(X2,1)=5             <172>
2352 IF(Y2=8)AND(F$="T")THEN A(X2,8)=8             <088>
2353 IF(Y2=1)AND(F$="T")THEN A(X2,1)=2             <158>
2354 IF(Y2=8)AND(F$="H")THEN A(X2,8)=10            <031>
2355 IF(Y2=1)AND(F$="H")THEN A(X2,1)=4             <170>
2356 IF(Y2=8)AND(F$="L")THEN A(X2,8)=9             <096>
2357 IF(Y2=1)AND(F$="L")THEN A(X2,1)=3             <166>
2370 X=X2:Y=Y2:GOSUB 2530                          <128>
2380 POKE 214,10:PRINT:PRINT TAB(28)CHR$(5
     );                                            <019>
2381 IF F$="D"THEN PRINT"{RVSON}DAME-"             <175>
2382 IF F$="T"THEN PRINT"{RVSON}TURM-"             <074>
2383 IF F$="H"THEN PRINT"{RVSON}SPRINGER-"         <051>
2384 IF F$="L"THEN PRINT"{RVSON}LAEUFER-"          <042>
2390 PRINT TAB(28)"{RVSON}TAUSCH"                  <153>
2400 GOSUB 2660:POKE 214,010:PRINT:PRINT T
     AB(28)"{RVSON,9SPACE}":GOTO 1730              <077>
2410 PRINT TAB(28)"{RVSON,4SPACE}":GOTO 17
     30                                            <235>
2420 REM ****** UNTERPROGRAMME ******              <081>
2430 REM                                           <206>
2440 REM -- EINZELNE FELDER --                     <160>
2450 REM -- DRUCKEN         --                     <186>
2460 REM                                           <236>
2470 IF B(X,Y)=1 THEN 2490                         <218>
2480 GOSUB 2840:POKE 646,15:PRINT"{RVSON}A
     AA{DOWN,3LEFT}AAA{DOWN,3LEFT}AAA":RET
     URN                                           <203>
2490 GOSUB 2840:PRINT"{3SPACE,DOWN,3LEFT,3
     SPACE,DOWN,3LEFT,3SPACE}":RETURN              <137>
2500 REM                                           <020>
2510 REM -- FIGUR DRUCKEN --                       <063>
2520 REM                                           <040>
2530 F=A(X,Y)                                      <146>
2540 GOSUB 2840                                    <218>
2550 POKE 646,9:IF F>6 AND B(X,Y)=0 THEN:P
     RINT"{RVSON}"A$(F-6)"{UP}":RETURN             <072>
```




```
2560 IF F>6 AND B(X,Y)=1 THEN:PRINT"{RVOFF
     }"A$(F-6)"{UP}":RETURN                   <239>
2570 POKE 646,8:IF B(X,Y)=1 THEN PRINT"{RV
     OFF}"A$(F)"{UP}":RETURN                  <099>
2580 PRINT"{RVSON}"A$(F)"{UP}":RETURN         <001>
2590 REM                                      <112>
2600 REM -- UMRECHNEN --                      <055>
2610 REM                                      <132>
2620 B1$=LEFT$(B$,1):X2=ASC(B1$)-64:Y2=VAL
     (RIGHT$(B$,1)):RETURN                    <224>
2630 REM                                      <152>
2640 REM -- ZEITSCHLAUFE --                   <029>
2650 REM                                      <172>
2660 POKE 198,0:FOR T=1 TO 250:IF PEEK(197
     )<>64 THEN 2680                          <002>
2670 NEXT:RETURN                              <069>
2680 IF PEEK(197)=3 THEN 2700                 <037>
2690 RETURN                                   <208>
2700 IF PEEK(197)<>64 THEN 2700               <106>
2710 IF PEEK(197)=64 THEN 2710                <219>
2720 RETURN                                   <238>
2730 REM                                      <252>
2740 REM -- GRUNDSTELLUNG --                  <191>
2750 REM                                      <016>
2760 A(1,1)=8:A(2,1)=10:A(3,1)=9:A(4,1)=11
     :A(5,1)=12:A(6,1)=9                      <204>
2770 A(7,1)=10:A(8,1)=8                       <244>
2780 FOR I=1 TO 8:A(I,8)=A(I,1)-6:NEXT        <123>
2790 FOR X=1 TO 8:A(X,2)=7:A(X,7)=1:NEXT      <141>
2800 RETURN                                   <062>
2810 REM                                      <076>
2820 REM -- BILDSCHIRMPOSITION --             <116>
2830 REM                                      <098>
2840 IF Y=8 THEN PRINT"{HOME}"TAB((X-1)*3+
     1);:RETURN                               <093>
2850 X1=(X-1)*3+1:Y1=(8-Y)*3-1:POKE 214,Y1
     :PRINT:PRINT TAB(X1);:RETURN             <010>
2860 REM                                      <128>
2870 REM -- SPIELENDE ---                     <069>
2880 REM                                      <148>
2890 POKE 198,0:FOR I=1 TO 10000:IF PEEK(1
     97)=64 THEN NEXT                         <204>
2900 FOR I=1 TO 8:FOR J=1 TO 8:A(I,J)=0:NE
     XT:NEXT                                  <252>
2910 GOSUB 2760:GOTO 1290                     <028>
10000 REM                                     <156>
10010 REM -----  DATA FUER DIE   ------       <233>
10020 REM -----  PARTIEN         ------       <009>
10030 REM                                     <186>
10040 REM ** DEMO-PARTIE 1 **                 <250>
10050 DATA SIFF,KASHDAN,1948                  <224>
10060 DATA D2D4,G8F6,C2C4,E7E6,B1C3,F8B4,D
      1C2,D7D5,A2A3,B4E7,C4D5,E6D5,C1F4,C7
      C6                                      <049>
10070 DATA H2H3,KR,E2E3,F8E8,F1D3,B8D7,G1F
      3,D7F8,F3E5,E7D6,KR,F8E6,F4H2,G7G6      <113>
10080 DATA E5F3,E6G7,H2D6,D8D6,F1C1,C8F5,F
      3D2,E8E7,B2B4,A8E8,A1B1,F5H3,G2H3,E7
      E3                                      <079>
10090 DATA D3F1,G7F5,F2E3,D6G3S,F1G2,G3E3S
      ,G1H1,F5G3S,H1H2,E3F4,D2F3,E8E2,C3E2    <224>
10100 DATA G3E2S,H2H1,F6H5,C2D2,H5G3S,H1H2
      ,G3F1S,H2H1,F4H2S,F3H2,F1G3M            <041>
10110 DATA E                                  <004>
10120 REM ** DEMO-PARTIE 2 **                 <075>
10130 DATA GELLER,EUWE,1953                   <082>
10140 DATA D2D4,G8F6,C2C4,E7E6,B1C3,F8B4,E
      2E3,C7C5,A2A3,B4C3,B2C3,B7B6,F1D3,C8
      B7                                      <240>
10150 DATA F2F3,B8C6,G1E2,KR,KR,C6A5,E3E4,
      F6E8,E2G3,C5D4,C3D4,A8C8,F3F4,A5C4      <140>
10160 DATA F4F5,F7F6,F1F4,B6B5,F4H4,D8B6,E
      4E5,C4E5,F5E6,E5D3,D1D3,B6E6,D3H7S      <195>
10170 DATA G8F7,C1H6,F8H8,H7H8,C8C2,A1C1,C
      2G2S,G1F1,E6B3,F1E1,B3F3,WA             <128>
10180 DATA E                                  <074>
10190 REM ** DEMO-PARTIE 3 **                 <146>
10200 DATA ROSSETTO,STAHLBERG,1947            <097>
10210 DATA E2E4,E7E6,D2D4,D7D5,B1C3,G8F6,C
      1G5,F8E7,E4E5,F6D7,H2H4,C7C5,C3B5,F7
      F6                                      <255>
10220 DATA F1D3,A7A6,D1H5S,E8F8,H1H3,A6B5,
      G5H6,D8A5S,H6D2,A5C7,H3G3,C5D4,G1F3     <249>
10230 DATA D7E5,G3G7,H7H6,D3H7,F8G7,H5H6S,
      G7F7,H6H5S,F7G7R                        <094>
10240 DATA X                                  <032>
```


Listing »Schachmeister«. (Schluß)

Alle Neune

Jeder Kegelverein macht mit viel Aufwand die statistische Auswertung der Kegelresultate. Mit dem Programm »Jahresauswertung« für den Commodore 64 bleibt Ihnen die Rechnerei erspart.

Das Programm »Jahresauswertung« stellt für jeden Teilnehmer eines Kegelvereins die Kegelergebnisse in einer Tabelle zusammen. Natürlich läßt sich die »Jahresauswertung« auch für Vereine mit ähnlicher Problematik anwenden, zum Beispiel Bowling-Gruppen, Schützenvereine oder Minigolf-Clubs. Die »ehrenvolle« Aufgabe der statistischen Auswertung aller Einzelergebnisse ist meist sehr mühsam. Was liegt da näher, als hierzu Hobby und notwendiges Übel miteinander zu verbinden und den Commodore arbeiten zu lassen. Neben der Zeitersparnis bei der Auswertung liegt für jedes Mitglied ein sauber gedrucktes Dokument vor (Bild 1), das zum Beispiel bei der Abschlußfeier überreicht werden kann.

Da die Ausdrücke beim Kegeln regional verschieden sind, hier einige Erklärungen dazu. Beim »Bilder-Kegeln« werden bestimmte Kegel-Konfigurationen aufgestellt, die mit einem Schub abzuräumen sind. »Pudel« oder »Ratten« heißen die Kugeln, die vorzeitig aus der Bahn und gegen die Bande laufen. »Stier« oder »Straße« sagt man, wenn der vordere Kegel, der König und der hintere Kegel bei einem Schub fallen.

Wollen Sie für Ihren Verein die Spalten anders benennen, so ändern Sie die Zeilen 1070 bis 1110. Sind Ihre Namen länger oder kürzer als die ursprünglichen, müssen Sie auch die Leerstellen zwischen den Wörtern berücksichtigen. Achten Sie darauf, daß die Namen teilweise über zwei Zeilen gehen. Also programmieren Sie erst die obere Reihe und im nächsten Schritt den unteren Teil.

Die wichtigsten Variablen

Zeile 230:	N	Anzahl der Spalten
Zeile 230:	M	Anzahl der Kegelabende
Zeile 250:	A	Anzahl der Anwesenheitstage
Zeile 270:	K$	Name des Keglers
Zeile 340:	U	Eingabewert für die Tabelle
Zeile 370:	D$	Datum des Kegelabends
Zeile 770:	S	Summe der Spalte
Zeile 790:	V	Durchschnitt der Spalte 1
Zeile 800:	B	Durchschnitt der Spalte 2
	I	Zähler für Anzahl der Kegelabende
	J	Zähler für Anzahl der Spalten

Feste Programmeinträge: Kalenderjahr: Zeilen 1510, 1700; Anzahl der Kegelabende: Zeilen 1520, 1770, 1810; DATA-Eintragungen der Abende: Zeilen 1770 bis 1810.
Gerätespezifische Angaben: Zeile 220: Bildschirmfarben; Zeile 1180: Kontrollton nach Abschluß der Eingaben eines Tages. Die Druckanweisungen gelten für Commodore-Drucker und müssen für andere Fabrikate gegebenenfalls entsprechend geändert werden.

(Jürgen Schäfer/ev)

```
100 REM ********************************   <231>
110 REM                                     <172>
120 REM   J A H R E S A U S W E R T U N G   <008>
130 REM                                     <192>
140 REM   (C)JUERGEN SCHAEFER , 1984        <082>
150 REM                                     <212>
160 REM   8000  M U E N C H E N  70         <022>
170 REM                                     <232>
180 REM         TEL.089-7601642             <147>
190 REM                                     <252>
200 REM ********************************    <250>
210 PRINT"{CLR}"                            <198>
220 POKE 53281,0:POKE 53280,0:POKE 646,1    <161>
225 REM---------------EINGABE ROUTINE-----
    -----------                             <106>
230 N=7:M=24                                <180>
240 PRINT"ANZAHL DER ANWESENHEITSTAGE IN 1
    984":PRINT                              <124>
250 INPUT A                                 <026>
260 IF A>M THEN PRINT"{CLR}":GOTO 240       <092>
270 PRINT:PRINT"GEBEN SIE DEN NAMEN DES KE
    GLERS EIN !":PRINT :INPUT K$            <223>
280 PRINT"{CLR}"                            <014>
290 PRINT"GEBEN SIE DIE EINZELWERTE DER SP
    ALTEN":PRINT                            <210>
300 PRINT"IN FOLGENDER REIHENFOLGE EIN :"   <238>
310 FOR T=1 TO 2000                         <195>
320 NEXT T                                  <238>
330 GOSUB 1160                              <240>
340 DIM U(M+1,N+1)                          <020>
350 FOR I=1 TO M                            <208>
360 GOSUB 1160                              <014>
370 READ D$                                 <246>
380 PRINT                                   <228>
390 PRINT D$                                <082>
400 PRINT"========":PRINT                   <211>
410 FOR J=1 TO N                            <149>
430 PRINT J;".SPALTE ";                     <158>
440 INPUT U(I,J):U$=STR$(U(I,J))            <082>
442 IF LEN(U$)>4 THEN 445                   <018>
444 GOTO 450                                <222>
445 PRINT: PRINT"BITTE NUR ZAHLEN BIS MAX.
     999 EINGEBEN !"                        <039>
446 PRINT"EINGABEWIEDERHOLUNG":PRINT :GOTO
     440                                    <100>
450 NEXT J                                  <032>
460 NEXT I                                  <034>
470 RESTORE                                 <010>
475 REM---------------HARDCOPY AUSDRUCK---
    -----------                             <164>
480 PRINT:INPUT"IST DER DRUCKER EINGESCHAL
    TET (J)";P$                             <211>
490 IF P$<>"J" THEN 480                     <039>
500 PRINT:PRINT"ES WIRD AUSGEDRUCKT"        <016>
520 GOSUB 1350                              <176>
530 GOSUB 1040                              <106>
540 OPEN 4,4                                <135>
550 REM---------------TABELLE ERZEUGEN----
    -----------                             <128>
560 FOR I=1 TO M                            <164>
570 READ D$                                 <192>
580 PRINT#4,D$;"{2SPACE}";                  <141>
590 FOR J=1 TO N                            <075>
600 X=U(I,J)                                <176>
610 IF X>99 THEN PRINT#4,X;:GOTO 640        <054>
620 IF X <100 AND X>=9  THEN PRINT#4,X;:GO
    TO 650                                  <194>
630 IF X <10 THEN PRINT#4,X;:GOTO 660       <041>
640 PRINT#4,    "{5SPACE}";:GOTO 670        <147>
650 PRINT#4,    "{6SPACE}";:GOTO 670        <157>
660 PRINT#4,    "{7SPACE}";:GOTO 670        <167>
670 NEXT J                                  <254>
680 PRINT#4                                 <175>
690 NEXT I                                  <010>
700 PRINT#4,"YYYYYYYYYYYYYYYYYYYYYYYYYYYYYY
    YYYYYYYY";                              <065>
710 PRINT#4,"YYYYYYYYYYYYYYYYYYYYYYYYYYYYYY
    YYYYYYYY"                               <152>
720 REM---------------SUMME PRO SPALTE BIL
    DEN--------                             <150>
730 PRINT#4,"SUMME:{3SPACE}";               <187>
740 FOR J=1 TO N                            <225>
750 S=0                                     <215>
760 FOR I=1 TO M                            <108>
770 S=S+U(I,J)                              <240>
780 NEXT I                                  <102>
790 IF J=1 THEN V=INT(S/A+.5)               <162>
```

```
800 IF J=2 THEN B=INT(S/A+.5)               <078>
810 PRINT#4,S;                              <045>
830 IF S>99 AND S<999 THEN GOTO 890         <120>
840 IF S<100 AND S>9 THEN GOTO 920          <193>
850 GOTO 950                                <160>
890 PRINT#4,"{5SPACE}";                     <174>
900 NEXT J                                  <230>
910 GOTO 970                                <252>
920 PRINT#4,"{6SPACE}";                     <204>
930 NEXT J                                  <004>
940 GOTO 970                                <026>
950 PRINT#4,"{7SPACE}";                     <234>
960 NEXT J                                  <034>
970 U(M+1,J)=S                              <241>
980 PRINT#4                                 <221>
990 PRINT#4:PRINT#4,"JAHRES-DURCHSCHNITT B
    EI 10X IN DIE VOLLEN :";CHR$(14);V;     <073>
1000 PRINT#4:PRINT#4,"JAHRES-DURSCHNITT BE
     IM BILDER-KEGELN{6SPACE}:";CHR$(14);B  <060>
1010 CLOSE 4                                <027>
1020 GOSUB 1620                             <118>
1030 END                                    <016>
1040 REM--------------TABELLENKOPF--------
     -----------                            <022>
1050 OPEN 4,4                               <137>
1060 PRINT#4,"KEGEL- ";" ";                 <113>
1070 PRINT#4,"10 SCHUB ";" ";" SUMME ";  "{
     3SPACE}";" PUDEL";                     <228>
1080 PRINT#4,"{4SPACE}";"STIERE";"{5SPACE}
     ";"ALLE ";"{4SPACE}";"KRANZ";"{4SPACE
     }";"SCHNAPS"                           <179>
1090 PRINT#4,"ABEND{2SPACE}";" ";           <227>
1100 PRINT#4,"I.D.VOLL.";" ";" BILDER ";"{
     2SPACE}";"(RATTEN)";                   <141>
1110 PRINT#4,"{2SPACE}";"{4SPACE}";"{7SPAC
     E}";"NEUN ";"{4SPACE}";"{5SPACE}";"{4
     SPACE}";"RUNDEN:"                      <248>
1120 PRINT#4,"YYYYYYYYYYYYYYYYYYYYYYYYYYYYY
     YYYYYYYY";                             <168>
1130 PRINT#4,"YYYYYYYYYYYYYYYYYYYYYYYYYYYYY
     YYYYYYYYYY"                            <014>
1140 CLOSE 4                                <159>
1150 RETURN                                 <192>
1160 REM--------------TON-----------------
     -----------                            <091>
1170 PRINT                                  <000>
1180 POKE 54273,23:POKE 54272,181           <137>
1190 POKE 54296,15                          <009>
1200 POKE 54277,9                           <078>
1210 POKE 54278,0                           <199>
1220 POKE 54276,33                          <162>
1230 FOR T=1 TO  500:NEXT T                 <099>
1240 POKE 54276,32                          <178>
1250 POKE 54273,0                           <108>
1260 PRINT"{CLR}"                           <232>
1265 PRINT"SPALTENBELEGUNG:"                <032>
1266 PRINT"YYYYYYYYYYYYYYYY"                <135>
1270 PRINT"1. 10 SCHUB IN DIE VOLLEN"       <228>
1280 PRINT"2. SUMME BILDERKEGELN "          <125>
1290 PRINT"3. PUDEL (RATTEN)"               <163>
1300 PRINT"4. STIERE(STRASSE)"              <063>
1310 PRINT"5. ALLE NEUN"                    <012>
1320 PRINT"6. KRANZ"                        <035>
1330 PRINT"7. SCHNAPSRUNDEN":PRINT:PRINT    <014>
1340 RETURN                                 <128>
1350 REM--------------FORMULARKOPF--------
     -----------                            <141>
1360 OPEN 4,4                               <193>
1370 PRINT#4,"*****************************
     ************";                         <131>
1380 PRINT#4,"*****************************
     ************" :PRINT#4:PRINT#4         <252>
1390 PRINT#4,CHR$(16)" 2";CHR$(14) CHR$(14
     )"JAHRESAUSWERTUNG FUER 1984{2SPACE}"
     ;                                      <032>
1400 PRINT#4:PRINT#4                        <072>
1410 PRINT#4,"UUUUUUUUUUUUUUUUUUUUUUUUUUUUU
     UUUUUUUUUUUU";                         <166>
1420 PRINT#4,"UUUUUUUUUUUUUUUUUUUUUUUUUUUUU
     UUUUUUUUUUUU"                          <019>
1430 PRINT#4,"KEGELCLUB ALLE NEUN ";        <031>
1440 PRINT#4,CHR$(16)"62";"17.DEZEMBER 198
     4"                                     <117>
1450 PRINT#4                                <183>
1460 PRINT#4,"DIE FOLGENDE TABELLE ZEIGT D
     IE PERSOENLICHEN LEISTUNGEN UNSERES "
     ;                                      <126>
```




```
1470 PRINT#4,"KEGELBRUDERS :"                 <204>
1480 PRINT#4                                  <213>
1490 PRINT#4:PRINT#4,CHR$(16)"25" CHR$(14)
     ;K$                                      <025>
1500 PRINT#4                                  <233>
1510 PRINT#4:PRINT#4,"WAEHREND DES ABGELAU
     FENEN JAHRES 1984."                      <211>
1520 PRINT#4,"IN DIESEM JAHR HATTEN WIR IN
     SGESAMT 24 KEGELABENDE.";                <035>
1530 PRINT#4,"SIE HABEN";A;"MAL TEILGENOMM
     EN"                                      <007>
1540 PRINT#4,"UND ERREICHTEN DAMIT ";INT(A
     /24 *100);" %{2SPACE}DER MAX. MOEGLIC
     HEN ";                                   <248>
1550 PRINT#4,"ANWESENHEITSZEIT !"             <216>
1560 PRINT#4                                  <039>
1570 PRINT#4,"I H R E{2SPACE}P E R S O E N
      L I C H E N{2SPACE}E R G E B N I S S
     E{2SPACE}:"                              <080>
1580 PRINT#4,"==============================
     ==============================="        <045>
1590 PRINT#4 :PRINT#4                         <008>
1600 CLOSE 4                                  <111>
1610 RETURN                                   <144>
1620 REM---------------SCHLUSSFORMEL-------
     -----------                              <181>
1630 OPEN 4,4                                 <209>
1640 PRINT#4:PRINT#4,"ŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪ
     ŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪ";                      <062>
1650 PRINT#4,"ŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪŪ
     ŪŪŪŪŪŪŪŪŪŪŪŪ"                            <251>
1660 PRINT#4:PRINT#4,"IM NAMEN DER GANZEN
     KEGELGRUPPE WUENSCHEN DER COMPUTER "; <238>
1670 PRINT#4,"UND DER VORSTAND IHNEN, "      <070>
1680 PRINT#4                                  <159>
1690 PRINT#4,"LIEBER HERR ";K$;",";           <243>
1700 PRINT#4,CHR$(14)"{3SPACE}GUT HOLZ FUE
     R 1985 !"                                <111>
1710 PRINT#4                                  <189>
1730 PRINT#4                                  <209>
1740 PRINT#4,"****************************
     ************";                           <247>
1750 PRINT#4,"****************************
     ************" :PRINT#4:PRINT#4           <112>
1760 CLOSE 4                                  <015>
1765 REM---------------DATEN DER KEGELABEND
     E-----------                             <044>
1770 DATA " 2.JAN.","16.JAN.","30.JAN.","1
     3.FEB.","27.FEB."                        <138>
1780 DATA "12.MAR.","26.MAR."," 9.APR.","
     7.MAI ","21.MAI "                        <240>
1790 DATA " 4.JUN.","18.JUN."," 2.JUL.","1
     6.JUL.","30.JUL."                        <004>
1800 DATA "13.AUG.","27.AUG.","10.SEP.","2
     4.SEP."," 8.OKT."                        <217>
1810 DATA "22.OKT."," 5.NOV.","19.NOV.","
     3.DEZ."                                  <042>
```

© 64'er

Listing zur »Jahresauswertung«. Bitte mit dem Checksummer V3 auf Seite 16 eingeben.

```
*****************************************************************************
J A H R E S A U S W E R T U N G   F U E R   1 9 8 4
```

KEGELCLUB ALLE NEUN 17.DEZEMBER 1984

DIE FOLGENDE TABELLE ZEIGT DIE PERSOENLICHEN LEISTUNGEN UNSERES KEGELBRUDERS :

PETER MUELLER

WAEHREND DES ABGELAUFENEN JAHRES 1984.
IN DIESEM JAHR HATTEN WIR INSGESAMT 24 KEGELABENDE.SIE HABEN 19 MAL TEILGENOMMEN
UND ERREICHTEN DAMIT 79 % DER MAX. MOEGLICHEN ANWESENHEITSZEIT !

I H R E P E R S O E N L I C H E N E R G E B N I S S E :

KEGEL- ABEND	10 SCHUB I.D.VOLL.	SUMME BILDER	PUDEL (RATTEN)	STIERE	ALLE NEUN	KRANZ	SCHNAPS RUNDEN:
2.JAN.	59	98	0	0	0	0	0
16.JAN.	51	78	3	1	1	0	0
30.JAN.	54	112	1	0	1	1	1
13.FEB.	48	114	1	0	0	1	0
27.FEB.	51	108	2	1	0	0	1
12.MAR.	54	123	0	0	1	0	1
26.MAR.	0	0	0	0	0	0	0
9.APR.	0	0	0	0	0	0	0
7.MAI	56	98	0	1	1	0	0
21.MAI	51	114	1	0	0	1	0
4.JUN.	0	0	0	0	0	0	0
18.JUN.	52	118	1	0	0	1	0
2.JUL.	55	124	1	0	1	1	1
16.JUL.	58	91	0	1	3	0	1
30.JUL.	45	99	1	2	0	0	0
13.AUG.	0	0	0	0	0	0	0
27.AUG.	49	101	0	0	1	0	1
10.SEP.	46	109	1	1	0	0	0
24.SEP.	51	112	1	1	0	1	1
8.OKT.	59	117	0	0	2	0	1
22.OKT.	0	0	0	0	0	0	0
5.NOV.	61	132	0	0	0	0	0
19.NOV.	53	137	2	0	0	1	0
3.DEZ.	50	98	1	0	0	0	1

SUMME: 1003 2083 16 8 11 7 9

JAHRES-DURCHSCHNITT BEI 10X IN DIE VOLLEN : 53
JAHRES-DURSCHNITT BEIM BILDER-KEGELN : 110

IM NAMEN DER GANZEN KEGELGRUPPE WUENSCHEN DER COMPUTER UND DER VORSTAND IHNEN,
LIEBER HERR PETER MUELLER, G U T H O L Z F U E R 1 9 8 5 !

```
*****************************************************************************
```

Bild 1. Beispielausdruck einer Auswertung

Weißt Du wieviel Sternlein stehen ...

Der C 64 als elektronische Sternenkarte bringt Klarheit ins Dunkel der Nacht. Das Programm zeigt Ihnen, wo Planeten und Sternbilder am Nachthimmel exakt stehen.

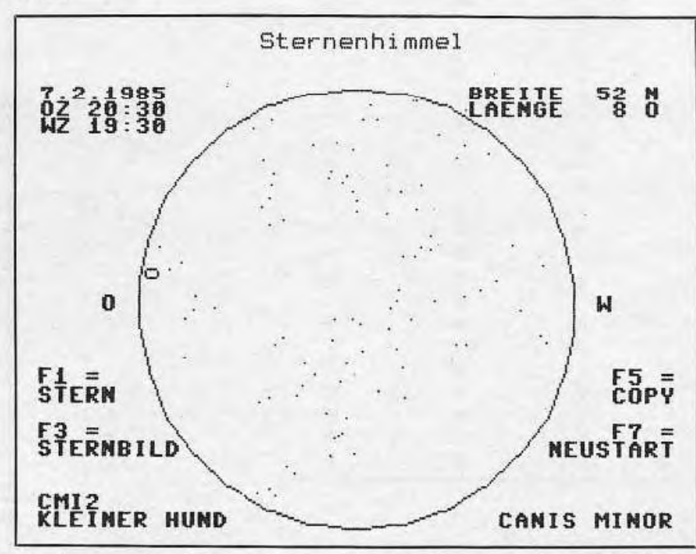
Das Programm entstand aus dem Bedürfnis, den Standort von Sternen, Sternbildern und Planeten zu ermitteln. Vor allem die Planeten lassen sich so ohne weiteres mit dem bloßen Auge, nicht erkennen. Man kann sie nur identifizieren, wenn man genaue Informationen über ihren jeweiligen Standort hat. Deshalb wurde die Berechnung der Planetenbahnen, der Bahn der Sonne und der des Mondes in das Programm aufgenommen. Das Programm erstellt eine Sternenkarte für einen eingegebenen Standort zum gewünschten Beobachtungszeitpunkt.

Die Sternenkarte enthält die hellsten und bekanntesten Fixsterne beziehungsweise Sternbilder des nördlichen Sternenhimmels.

Sehr interessant ist es, neben der Ausgabe der Sternenkarte für eigene Beobachtungen, sich die Sternenkarte für jeden Punkt der Erde zeichnen zu lassen.

So kann man sich zum Beispiel in Gedanken auf den Nordpol versetzen. Der Polarstern (Polaris) steht im Zenit, also genau über dem Beobachter. Gibt man als Beobachtungszeitpunkt den 21. März (Frühlingsanfang für die Nordhalbkugel) ein, kann man feststellen, daß die Sonne den ganzen Tag über scheint.

Am Nordpol hat der Tag begonnen. Die Sonne scheint nun ein halbes Jahr ohne Unterbrechung. Erst zum Herbstanfang geht sie wieder unter. Es ist dann für ein halbes Jahr Nacht. Auch die Verhältnisse am Äquator (Breite 0°) sind sehr interessant.



Hardcopy des Sternenhimmels von Hamm (zirka 52° nördliche Breite und 8° östliche Länge), wobei der Beobachtungszeitpunkt der 7. Februar 1985, 19.30 GMT (Weltzeit) beziehungsweise 20.30 MEZ (Ortszeit) ist. Wie bei Sternkarten üblich, ist Osten links, Westen rechts, Norden oben und Süden unten. Der Mond steht tief im Osten.

Nach dem Start des Programms werden zunächst das aktuelle Datum sowie die aktuelle Uhrzeit (MEZ) und die geografischen Koordinaten des Beobachtungsortes eingegeben. Dabei müssen nördliche Breiten und westliche Längen positiv, südliche Breiten und östliche Längen negativ angegeben werden. Nun kann man den Beobachtungszeitpunkt wählen, wenn sich dieser vom bereits im Programm festgelegten Zeitpunkt unterscheidet.

Es folgt eine kurze Erläuterung des Programmablaufs und das Zeichnen der Sternkarte. Mit den Cursor-Tasten kann über ein Kreuz jedes Objekt der Sternkarte angesteuert werden. Die Bewegungsgeschwindigkeit ist abhängig von der Dauer des Tastendrucks; das heißt je länger eine Cursortaste gedrückt wird, desto schneller bewegt sich das Kreuz. Drückt man nach der Wahl die Funktionstaste F1, gibt der Computer den Namen des Sterns oder Planeten, der dem Kreuz am nächsten ist, an. Dabei wird der Cursor genau auf das Objekt positioniert. Möchte man wissen, um welches Sternbild es sich bei einer bestimmten Konstellation handelt, bringt man das Kreuz auf das betreffende Sternbild und drückt die Funktionstaste F3. Der Computer positioniert den Cursor auf den nächsten Stern des Sternbildes und zeigt den Namen des Sternes und des Sternbildes an. Das gesamte Sternbild blinkt. Es besteht auch die Möglichkeit, einen Stern, ein Sternbild oder einen Planeten nach dem Namen suchen zu lassen. Dazu gibt man einfach mit der Tastatur den Namen oder einen Teil des Namens des gesuchten Objekts ein und schließt die Eingabe mit der RETURN-Taste ab. Der Computer positioniert das Kreuz auf dieses Objekt und gibt den vollständigen Namen aus. Ist das gesuchte Objekt nicht gespeichert oder zum Beobachtungszeitpunkt am Beobachtungsort nicht zu sehen, erhält man eine entsprechende Fehlermeldung.

Will man die Sternkarte für einen anderen Beobachtungsort oder einen anderen Beobachtungszeitpunkt erstellen lassen, drückt man die Funktionstaste F7.

Mit F5 wird die Sternkarte auf einen MPS 801 gedruckt.

Programmerläuterung

Auf die Formeln, die den Berechnungen zugrunde liegen, soll verzichtet werden. Diese Berechnungen sind (zumindest für Mond- und Planetenpositionen) ziemlich kompliziert und können bei Interesse jedem größeren Werk über Astronomie entnommen werden.

Von programmtechnischem Interesse dürften allerdings die Suchroutinen sein. Deren Funktionsweise soll deshalb erläutert werden.

Zum Suchen werden die folgenden Arrays verwendet:

P$	Namen der Planeten
Z$	Namen der Sternbilder
P	Bildschirmkoordinaten der Planeten
Z%	Bildschirmkoordinaten der Fixsterne
Z	Zeiger auf Fixsterne (für jedes Sternbild)

Suchen Fixstern/Planet (Funktionstaste F1)

Zunächst werden die Bildschirmkoordinaten des Kreuzes (X,Y) mit denen der Planeten (X mit P(0), P(2), ..., Y mit P(1), P(3), ...) verglichen. Bei Übereinstimmung ist XX=0 und die Suche kann abgebrochen werden. In Zeile 42090 verzweigt das Programm nach 45000, wo der Name des Planeten geschrieben wird. Liegt keine Übereinstimmung vor, wird der Planet mit dem geringsten Abstand gespeichert (XP = Abstand^2, ZP = Kennziffer des Planeten). Nun werden die Fixsterne in gleicher Weise überprüft. Der Stern mit dem geringsten Abstand wird gespeichert (Z = Kennziffer des Sterns; Zeile 44020). Wird kein Stern mit geringerem Abstand als der nächste Planet gefunden, wird der gespeicherte Planet genannt (Zeilen 44070 bis 45010). Dabei wird ab der Zeile 500 das Kreuz exakt auf den Planeten beziehungsweise Stern positioniert und dessen Bezeichnungen ausgegeben. Der Name eines Planeten ergibt sich aus dem Array P$ mit

Programmstruktur von Sternenhimmel

Zeilen	Bedeutung
100 - 120	Berechnung des Stundenwinkels
130 - 140	Berechnung Bogenmaß
150 - 190	Koordinaten-Transformation
200 - 240	Berechnung Gradmaß
296 - 395	Text auf Grafikbildschirm überschreiben
396 - 480	Ausgabe Fehlermeldungen
496 - 520	Kreuz (Cursor) bewegen
596 - 630	Buchstaben auf Grafikbildschirm löschen
996 - 2190	Bildschirmmaske erstellen
4996 - 5260	Definition Kreuz als Sprite
9996 - 15080	Stellung Erde/Sonne/Mond berechnen
19996 - 20160	Stellung Planeten berechnen
24996 - 25080	Stellung Fixsterne berechnen
29996 - 30050	Rektaszension und Deklination berechnen
39996 - 49100	Eingabeschleife
49996 - 56050	Programmbeginn und Funktionsdefinitionen:
52010 - 52020	arc sin
52030 - 52040	arc cos
52050 - 52060	Modulo 360
57000 - 57110	Hardcopy
58000 - 58020	Neustart des Programms
60000 - 61240	Daten Fixsterne (Rektaszension, Deklination, Name, Nummer des Sternbildes)
62000 - 62230	Daten Sternbilder (Name lateinisch, Name deutsch, Zahl Sterne —1, Beginn der Sterndaten DATA-Zeile 60000 + X·10)
62500 - 62700	Daten Planeten (Name, Bahnelemente)

dem Index ZP. Der Name eines Sternes wird durch berechnetes RESET (Zeile 44080) aus den DATA-Zeilen 60000 bis 61240 ermittelt. Die Schreibroutine befindet sich ab Zeile 300.

Suchen Sternbild (Funktionstaste F3)

Zunächst wird die Suchroutine für Sterne mit Ausnahme des Abschnitts für die Planeten durchlaufen. Der gefundene nächste Stern bestimmt das Sternbild. Die Kennziffer Z des Sterns wird in Z1, die letzte Ziffer des gelesenen Datensatzes für den Stern (D; Zeile 44090) wird in Z gespeichert. Das Programm verzweigt nun in die Blink-Routine (Zeile 48060 bis 48250). Dort werden die zum Sternbild gehörenden Sterne (Punkte) abwechselnd gesetzt und gelöscht (Zeile 48170).

Suchen nach Namen

Wird ein Name, oder ein Teil eines Namens eingegeben, verzweigt das Programm in 40100 nach Zeile 47000. Das komplizierte Löschen des zu überschreibenden Textes (GOSUB 340, GOSUB 360 in Zeile 47030) ist erforderlich, da Simons Basic Text mit den auf dem Grafik-Bildschirm gesetzten Punkten ODER-verknüpft. Aus diesem Grund wird der vorige Text erst gelöscht. Nun wird geprüft, ob ein Planetenname mit der eingegebenen Bezeichnung übereinstimmt (47060 bis 47090). Ist dies der Fall, werden die Bildschirmkoordinaten des Planeten zu denen des Kreuzes (Zeile 47120). Ist der Planet zur Beobachtungszeit nicht sichtbar, dann ist X = 0 und eine Fehlermeldung wird ausgegeben (47140). Ansonsten wird das Kreuz auf den Planeten positioniert und in die Eingaberoutine gesprungen (47150). Handelt es sich bei der eingegebenen Bezeichnung um keinen Planeten, wird die Suchroutine in 48000 fortgesetzt. Dort werden zunächst die Namen der Sternbilder (deutsch und lateinisch) auf Übereinstimmung untersucht. Wird keine Übereinstimmung gefunden, springt das Programm in die Zeile 49000. Hier werden nun die Sternbezeichnungen eingelesen und mit der Eingabe verglichen. Kommt es zu keiner Übereinstimmung, wird die Fehlermeldung »... nicht gespeichert« ausgegeben und das Kreuz in die linke obere Ecke des Bildschirms gesetzt (Zeile 49050). Darauf kehrt das Programm in die Eingaberoutine (ab Zeile 40000) zurück.

(Horst Hinkelmann/hm)

```
0 rem ********************************
1 rem *                              *
2 rem *        sternenhimmel         *
3 rem *                              *
4 rem *      horst hinkelmann        *
5 rem *                              *
8 rem *                              *
9 rem ********************************
10 goto50000
96 rem **********************
97 rem *    stern setzen    *
98 rem **********************
100 rem ** stundenwinkel **
120 sw=fnmo(ar-re-1)
130 rem ** bogenmass **
140 sw=sw*p1
150 rem ** koordinaten-transformation **
160 h=fnas(sb*sin(de)+cb*cos(de)*cos(sw)
)
170 ifh<0thenx=0:y=0:return
180 a=fnac((sin(de)-sb*sin(h))/(cb*cos(h
)))
190 ifsw<pithena=-a
200 rem ** gradmass **
210 r=99-h*198/pi:w=-a-pi/2
220 x=int(cos(w)*r*1.1+159.5)
230 y=int(sin(w)*r+99.5)
240 plotx,y,1
296 rem ***********************
297 rem * text ueberschreiben *
298 rem ***********************
300 n$=left$(n$,13)
310 ifn3$<>n$thengosub340
320 ifn4$<>n1$thengosub360
330 return
340 n3$=n$:block0,184,103,191,0
350 text0,184,n$,1,0,8:return
360 n4$=n1$:block0,192,119,199,0
370 block216,192,319,199,0
380 text0,192,n1$,1,0,8
390 text319-8*len(n2$),192,n2$,1,0,8
395 return
396 rem ***********************
397 rem *   fehlermeldungen   *
398 rem ***********************
400 n1$=left$(n1$,9)
410 text0,192,n1$+" nicht",1,0,8
420 text232,192,"sichtbar",1,0,8
430 goto41000
450 n$=left$(n$,9)
460 text0,192,n$+" nicht",1,0,8
470 text232,192,"gespeichert",1,0,8
480 goto41000
496 rem ***********************
497 rem *    kreuz bewegen    *
498 rem ***********************
500 hn=x+19:vn=y+46
510 mmob1,hv,vv,hn,vn,0,100
520 goto300
596 rem ***********************
597 rem * buchstaben loeschen *
598 rem ***********************
600 le=len(n$)-1
610 blockle*8,184,le*8+7,199,0
620 n$=left$(n$,le)
630 return
996 rem ************************
997 rem * bildschirm erstellen *
998 rem ************************
1000 hires7,6
1010 circle160,100,110,100,1
1020 hu=0:ifb<0thenhu=200
1030 ifb<0thencircle160,hu+b/9*10,2,2,1
1040 text1,0,mid$(str$(t),2)+"."+mid$(st
r$(m),2)+"."+mid$(str$(j),2),1,0,8
1050 text1,8,"oz"+oz$,1,0,8
1060 text1,16,"wz"+wz$,1,0,8
1070 t$=mid$(str$(int(b+.5)),2,4)
1080 sp$=right$("   ",3-len(t$))
```

Listing zu Sternenhimmel. Geben Sie das Listing nur dann ein, wenn Sie vorher Simons Basic geladen und aktiviert haben.

```
1090 text216,0,"breite "+sp$+t$,1,0,8
1100 t$=mid$(str$(int(l+.5)),2,4)
1110 sp$=right$("   ",3-len(t$))
1120 b$="n":ifb<0thenb$="s"
1130 text304,0,b$,1,0,8
1140 text216,8,"laenge "+sp$+t$,1,0,8
1150 l$="w":ifl<0thenl$="o"
1160 text304,8,l$,1,0,8
1170 text32,96,"o",1,0,8
1180 text280,96,"w",1,0,8
1190 gosub2010
1200 gosub2110
1210 text288,128,"f5 =",1,0,8
1220 text288,136,"copy",1,0,8
1230 text288,152,"f7 =",1,0,8
1240 text256,160,"neustart",1,0,8
1250 return
2000 block0,128,39,143,0
2010 text0,128,"f1 =",1,0,8
2020 text0,136,"stern",1,0,8
2030 return
2050 block0,128,39,143,0
2060 text0,128,"%sucht",1,0,8
2070 text0,136,"%stern",1,0,8
2080 return
2100 block0,152,39,159,0
2110 block0,160,71,167,0
2120 text0,152,"f3 =",1,0,8
2130 text0,160,"sternbild",1,0,8
2140 return
2150 block0,152,39,159,0
2160 block0,160,71,167,0
2170 text0,152,"%sucht",1,0,8
2180 text0,160,"%sternbild",1,0,8
2190 return
4996 rem **********************
4997 rem *    hires kreuz      *
4998 rem **********************
5000 design 0,32*64+49152
5020 @.....b..................
5030 @.....b..................
5040 @.....b..................
5050 @.....b..................
5060 @b.b.b.b.b.b.............
5070 @.....b..................
5080 @.....b..................
5090 @.....b..................
5100 @.....b..................
5110 @........................
5120 @........................
5130 @........................
5140 @........................
5150 @........................
5160 @........................
5170 @........................
5180 @........................
5190 @........................
5200 @........................
5210 @........................
5220 @........................
5230 mob set 1,32,0,1,0
5240 hn=179:vn=145
5250 mmob1,hv,vv,hn,vn,0,0
5260 return
9996 rem *********************
9997 rem *  erde/sonne/mond  *
9998 rem *********************
10000 rem ** erde **
10100 be=fnmo(tg*.985609121+99.18)
10200 el=fnmo(be+sin((be-102.2)*p1)*1.84
5)
10300 ea=1+sin((el-192.2)*p1)*.0167
11000 rem ** sonne **
11200 ls=fnmo(el+180)
12000 rem ** mond **
12010 lm=tg*13.1763976+51.23
12020 pm=tg*.111399014+208.9
12030 km=372.1-tg*.052953643
12040 lm=fnmo(lm)
12050 pm=fnmo(pm)
12060 km=fnmo(km)
12070 am=lm-pm
12080 km=km-sin(as*p1)*.16
12090 ms=(lm-ls)*2-am
12100 am=am+sin(ms*p1)*1.27388889-sin(as
*p1)*(.18638889+.36)
12110 lm=lm+sin(ms*p1)*1.27388889-sin(as
*p1)*.18638889+sin(am*p1)*6.28833333
12120 m1=lm-ls
12130 lm=lm+sin(m1*2*p1)*.658333333
12140 m2=lm-km
12150 lm=lm-sin(m2*2*p1)*.12
12160 bm=sin(m2*p1)*5.14539
12170 m3=(lm-ls)*2-m2
12180 bm=bm+sin(m3*p1)*.15
15000 n$="sonne":al=ls:ab=0:gosub30000
15010 ifx=0goto15040
15020 circlex,y,4,3,1
15030 paintx+1,y,1
15040 n$="mond":al=lm:ab=bm:gosub30000
15050 ifx=0goto15080
15060 plotx,y,0
15070 circlex,y,4,3,1
15080 return
19996 rem *********************
19997 rem *     planeten      *
19998 rem *********************
20000 reset62500
20010 forz=0to5
20020 readn$,tb,ep,ph,mp,e,kn,i,ae
20030 p$(z)=n$
20040 ml=fnmo(tb*tg+ep)
20050 wl=ml+sin((ml-ph)*p1)*mp
20060 sp=ae+sin((wl-ph-90)*p1)*e*ae
20070 ws=fnmo(c+el-wl)*p1:si=sin(ws)
20080 fl=ea/sp-cos(ws)
20090 we=atn(si/fl)*p2
20100 al=fnmo(el+we-180*(fl>=0))
20110 wt=sin((wl-kn)*p1)*i
20120 ab=atn(tan(wt*p1)*abs(sin(we*p1)/s
i))*p2
20130 gosub30000
20140 p(2*z)=x:p(2*z+1)=y
20150 next
20160 return
24996 rem *********************
24997 rem *    fixsterne      *
24998 rem *********************
25000 reset60000
25010 fors=1toi1
25020 readre,de,n$,a
25030 de=de*p1
25040 n1$=z$(a,1):n2$=z$(a,0)
25050 gosub100
25060 z%(s)=x:z%(s+i2)=y
25070 next
25080 return
29996 rem *********************
29997 rem *  rektas./deklin.  *
29998 rem *********************
30000 sn=sin(ab*p1):cs=cos(ab*p1)
30010 sl=sin(al*p1):cl=cos(al*p1)
30020 de=fnas(ec*sn+es*cs*sl)
30030 re=2*atn((ec*cs*sl-es*sn)/(cos(de)
+cs*cl))
30040 re=fnmo(re*p2)
30050 goto100
39996 rem *********************
39997 rem *    eingaben       *
39998 rem *********************
40000 getg$:ifg$=""thenv=1:goto40000
40010 p=asc(g$)
40020 ifp=17thenvn=vn+v:goto41000
40030 ifp=145thenvn=vn-v:goto41000
40040 ifp=29thenhn=hn+v:goto41000
40050 ifp=157thenhn=hn-v:goto41000
40060 ifp=133thengosub2050:goto42000
40070 ifp=134thengosub2150:goto42000
40080 ifp=135goto57000
40090 ifp=136goto58000
40100 ifp>64andp<91goto47000
40110 goto40000
41000 v=v+.5
41010 mmob1,hn,vn,hn,vn,0,0
41020 goto40000
42000 x=int(hn-19):y=int(vn-46)
42010 xm=200000
42020 ifp=134goto44000
```

Listing zu Sternenhimmel. (Fortsetzung)

```
42030 fora=0to10step2
42035 f=p(a)-x:h=p(a+1)-y
42040 xx=f*f+h*h
42050 ifxx<xmthenxm=xx:zp=a
42060 ifxx=0thena=12
42070 next
42080 n$=p$(zp/2):n1$=" ":n2$=" "
42090 ifxx=0goto45000
44000 xp=xm
44010 fora=0toi1
44015 f=z%(a)-x:h=z%(a+i2)-y
44020 xx=f*f+h*h
44030 ifxx<xmthenxm=xx:z=a
44040 ifxx=0thena=i2
44050 next
44060 x=p(zp):y=p(zp+1)
44070 ifxp=xmgoto45000
44080 x=z%(z):y=z%(z+i2)
44090 reset60000+z*10
44100 readre,de,n$,d
44110 n1$=z$(d,1):n2$=z$(d,0)
44120 ifp=134thenz1=z:z=d:gosub2100:goto
48060
45000 gosub2000
45010 hv=hn:vv=vn:gosub500:goto40000
47000 n$=g$:n1$=" ":n2$=" "
47010 gosub340:gosub360
47020 getg$:ifg$=""goto47020
47030 ifasc(g$)=20thengosub600:goto47020
47040 ifasc(g$)=13goto47060
47050 n$=n$+g$:gosub350:goto47020
47060 p=0:z=-1
47070 fors=0to5
47080 ifplace(n$,p$(s))thenz=s:s=5
47090 next
47100 ifz<0goto48000
47110 hv=hn:vv=vn
47120 x=p(2*z):y=p(2*z+1)
47130 n1$=p$(z):n2$=" "
47140 ifx=0goto400
47150 gosub500:goto40000
48000 fors=0toi3
48010 ifplace(n$,z$(s,0))thenz=s:s=i3
48020 ifplace(n$,z$(s,1))thenz=s:s=i3
48030 next
48040 ifz<0goto49000
48050 z1=-1
48060 a1=z(z,1):a2=a1+z(z,0)
48070 ifz1>=0goto48120
48080 fora=a1toa2
48090 ifz%(a)thenz1=a:a=a2
48100 next
48110 ifz1<0thenn1$=z$(z,1):goto400
48120 f=0:hv=hn:vv=vn
48130 n1$=z$(z,1):n2$=z$(z,0)
48140 x=z%(z1):y=z%(z1+i2):gosub500
48150 fora=a1toa2
48160 x=z%(a)
48170 ifxthenplotx,z%(a+i2),f
48180 next
48190 f=1-f
48200 iff=1goto48150
48210 getg$:p=asc(g$+chr$(0))
48220 ifpgoto40020
48240 fora=1to1000*f+10:next
48250 goto48150
49000 reset60000
49010 fors=1toi1
49020 readre,de,s$,a
49030 ifplace(n$,s$)thenz=s:s=i2
49040 next
49050 ifz<0thenhn=x+19:vn=y+46:goto450
49060 ifz%(z)=0thenn1$=s$:goto400
49070 hv=hn:vv=vn
49080 x=z%(z):y=z%(z+i2)
49090 n$=s$:n1$=z$(a,1):n2$=z$(a,0)
49100 gosub500:goto40000
49996 rem ********************
49997 rem *  programmbeginn  *
49998 rem ********************
50000 cset1
50010 ifpeek(900)>0andpeek(900)<11then50
110
50020 print"Heutiges Datum"
50030 input"Jahr ";j$
50035 j$="19"+right$(j$,2)
50040 input"Monat";m$
50050 input"Tag  ";t$
50060 a$=t$+"."+m$+"."+j$
50070 fora=1tolen(a$)
50080 poke900+a,asc(mid$(a$,a,1))
50090 poke900,a
50100 next
50110 t$=left$(ti$,4)
50120 print"Uhrzeit HHMM"
50130 printtab(8)t$
50140 print""tab(6);:inputt1$:
50150 ift$<>t1$thenti$=right$("0"+t1$+"0
0",6)
50160 a$=""
50170 fora=0topeek(900)
50180 a$=a$+chr$(peek(900+a))
50190 next
50200 i1=124:i2=125:i3=22:f=1
50210 pi=π:p1=pi/180:p2=180/pi:c=360
50220 dimz$(i3,1),z(i3,1),p(11),z%(2*i2)
50300 print""
50301 printtab(11)"┌────────────────┐"
50302 printtab(11)"│ Sternenhimmel  │"
50304 printtab(11)"└────────────────┘"
50310 print"Breite  90 (noerdl) bis  -9
0 (suedl.)"
50320 print"Laenge 180 (westl.) bis -180
 (oestl.)"
50330 print"Vorgegeben sind die Werte v
on Hamm und  die heutige Zeit."
50340 print"Breite      "52:print""tab(1
0);:inputb
50350 ifabs(b)>=90then50340
50360 print"Laenge      ";-8:print""tab
(10);:inputl
50370 ifabs(l)>180then50360
50380 sb=sin(b*p1):cb=cos(b*p1)
50390 print"Jahr         "right$(a$,4)
50400 print""tab(10);:inputj
50410 t=int(val(mid$(a$,2)))
50420 m=int(val(mid$(a$,len(str$(t))+2,2
)))
50430 print"Monat        "m
50440 print""tab(10);:inputm
50450 ifm>12orm<1then50440
50460 print"Tag          ";t
50470 print""tab(10);:inputt
50480 ift<1ort>31then50470
50490 wz=int(val(ti$)/100)-100
50495 ifwz<100thenwz=wz+2400
50500 print"Weltzeit    hhmm"
50505 printtab(12)right$(str$(wz),4)
50510 print""tab(10);:inputwz$:wz=val(w
z$)/100
50520 ifwz<0orwz>24goto50510
50530 zt=int(wz)+frac(wz)/.6
50600 i=m<3
50610 k=t+int((153*m-11*i-162)/5)+int((1
461*j+i)/4)+(j>=0)*366
50620 ifk>577736thenk=k-int((int((j+i)/1
00)*3-5)/4)
50630 ta=k-693596:tg=k-711858+zt/24:i=ta
/36525
50640 e=23.452294-i*.013125-i*i*1.639e-6
+i*i*i*5.028e-7
50650 es=sin(e*p1):ec=cos(e*p1)
52000 rem ** funktionen definieren **
52010 rem **  arcsin  **
52020 deffnas(x)=atn(x/sqr(1-x*x))
52030 rem **  arccos  **
52040 deffnac(x)=pi/2-atn(x/sqr(1-x*x))
52050 rem **  modulo  **
52060 deffnmo(x)=x-int(x/c)*c
54000 rem ** zeit **
54010 wz$=mid$(str$(int(frac(wz)*100+.5)
),2)
54020 wz$=right$("0"+wz$,2)
54030 wz$=right$("  "+str$(int(wz)),3)+":
"+wz$
54040 lo=int(l/15)*15
```

Listing zu Sternenhimmel. (Fortsetzung)

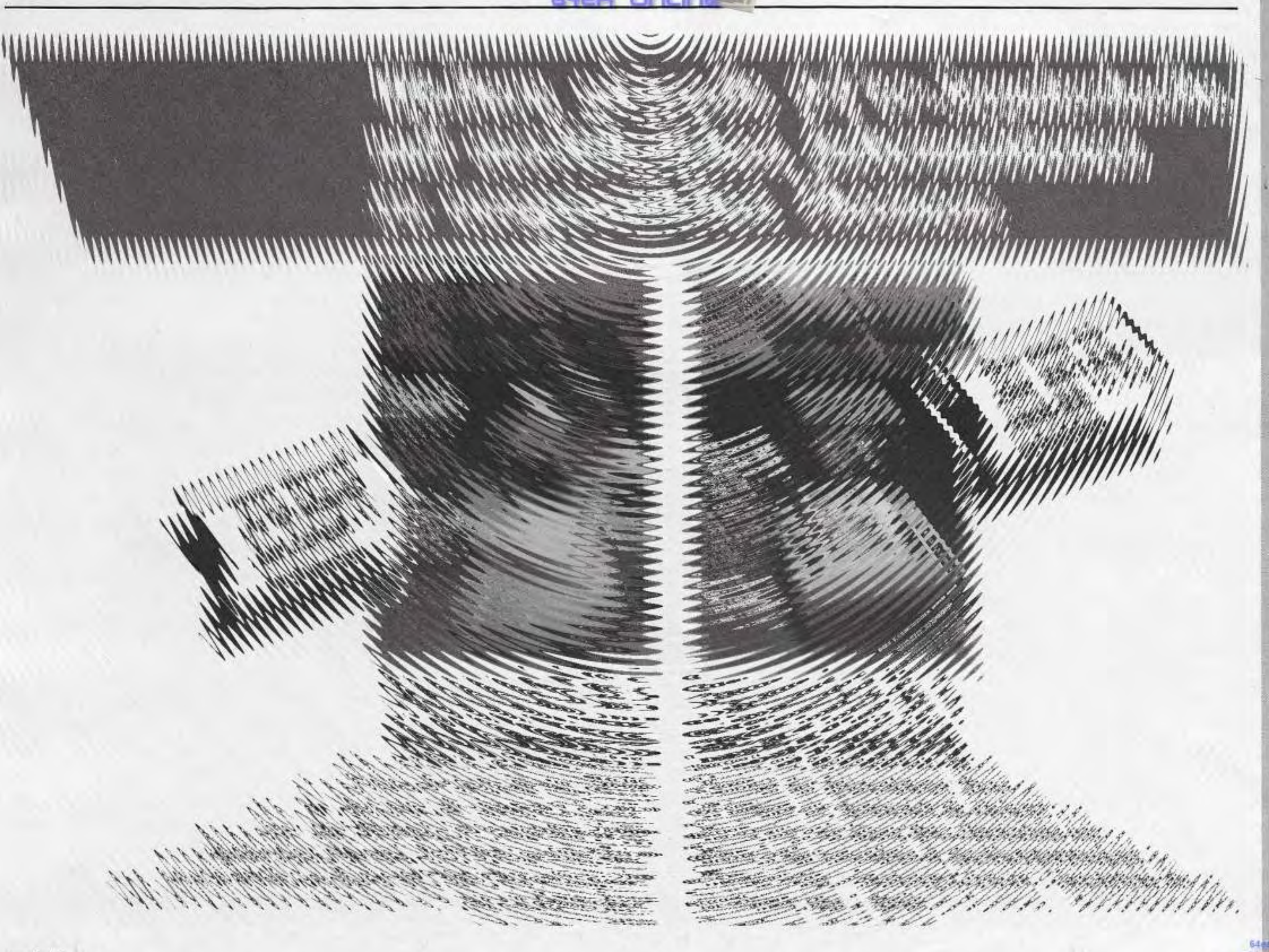
```
54050 oz=fnmo(zt*15-lo)/15
54060 oz$=right$(" "+str$(int(oz)),3)+ri
ght$(wz$,3)
54200 rem aries
54210 ar=zt*360.985647/24+frac(ta/1461)*
1440.02509
54220 ar=ar+int(ta/1461)*.0307572+99.201
8973
54230 ar=fnmo(ar)
54300 rem daten sternbilder
54310 reset62000
54320 fora=0toi3
54330 readz$(a,0),z$(a,1),z(a,0),z(a,1)
54340 next
54350 ifwhgoto56000
55000 rem ** anleitung **
55010 print"▒▒Dieses Programm zeichnet e
ine Sternen-"
55020 print"karte mit der Sonne, dem Mon
d, den"
55030 print"Planeten und den Fixsternen.
":print
55040 print"Die Namen der Sterne oder de
ren"
55050 print"Standorte koennen wie folgt
gesucht"
55060 print"werden:":print
55080 print"Mit den Cursor-Tasten das Kr
euz auf"
55090 print"oder in die Naehe des Sterne
s fuehren"
55100 print"und f1 druecken.":print
55110 print"Wenn das ganze Sternzeichen
gesucht"
55120 print"wird, dann f3 druecken.":pri
nt
55130 print"Wenn der Name oder ein Teil
des Namens"
55140 print"eines Sterns oder Sternbilde
s eingegeben";
55150 print"wird, geht das Kreuz alleine
 auf den":print"Stern"
55160 print"Bei einem Sternbild blinken
die dazu-"
55170 print"gehoerigen Sterne.":print
55180 print"Weiter = Taste druecken!";
55200 poke198,0:wait198,1
56000 gosub 1000:rem bildschirm
56010 gosub10000:rem sonne mond
56020 gosub20000:rem planeten
56030 gosub25000:rem fixsterne
56040 gosub 5000:rem kreuz
56050 goto40000
57000 rem ** hardcopy **
57100 copy
57110 goto40000
58000 rem ** neustart **
58010 nrm:cset1:mob off 1:wh=1
58020 goto50300
60000 rem ** daten fixsterne **
60010 data037.8,89.3,polaris,0
60020 data269.8,86.0,umi2,0
60030 data252.5,82.2,umi3,0
60040 data246.3,75.5,umi4,0
60050 data238.5,78.0,umi5,0
60060 data230.2,71.8,pherkad,0
60070 data222.7,74.2,kochab,0
60080 data206.9,49.3,benetnasch,1
60090 data201.0,54.9,mizar,1
60100 data193.5,56.0,alioth,1
60110 data183.9,57.0,megrez,1
60120 data178.5,53.7,phekda,1
60130 data165.9,61.8,dubhe,1
60140 data165.5,56.4,merak,1
60150 data28.6,63.7,cas1,2
60160 data21.5,60.2,cas2,2
60170 data14.2,60.7,cas3,2
60180 data10.1,56.5,schedir,2
60190 data2.3,59.2,caph,2
60200 data  3.3,15.2,algenib,3
60210 data346.2,15.2,markab,3
60220 data345.9,28.1,scheat,3
60230 data311.6,34.0,cyg1,4
60240 data310.4,45.3,deneb,4
60250 data305.6,40.3,schedir,4
60260 data296.2,45.1,cyg4,4
60270 data292.7,28.0,albireo,4
60280 data302.8,-0.8,aql1,5
60290 data297.7,08.9,atair,5
60300 data296.6,10.6,aql3,5
60310 data292.0,03.5,aql4,5
60320 data286.6,-4.9,aql5,5
60330 data286.4,13.9,aql6,5
60340 data284.7,32.7,lyr1,6
60350 data283.0,37.0,lyr2,6
60360 data282.5,33.4,lyr3,6
60370 data280.2,38.8,lyr4,6
60380 data279.2,38.8,wega,6
60390 data265.6,-39.0,sco1,7
60400 data264.3,-43.0,sco2,7
60410 data263.4,-37.1,sco3,7
60420 data262.7,-37.3,sco4,7
60430 data252.5,-34.3,sco5,7
60440 data249.0,-28.2,sco6,7
60450 data247.3,-26.4,antares,7
60460 data241.4,-19.8,acrab,7
60470 data240.1,-22.6,sco9,7
60480 data239.7,-26.1,sco10,7
60490 data228.9,33.3,boo1,8
60500 data225.0,41.0,boo2,8
60510 data221.2,27.1,boo3,8
60520 data218.0,38.3,boo4,8
60530 data218.0,30.5,boo5,8
60540 data213.9,19.2,arktur,8
60550 data201.3,-11.2,spika,9
60560 data198.0, -7.5,vir2,9
60570 data195.5, 11.0,vir3,9
60580 data193.9,  3.4,vir4,9
60590 data190.0, -1.0,vir5,9
60600 data185.0,  0.0,vir6,9
60610 data177.0,  2.0,vir7,9
60620 data191.9,-59.7,cru1,10
60630 data187.8,-57.1,cru2,10
60640 data186.7,-63.1,cru3,10
60650 data183.8,-58.8,cru4,10
60660 data177.3,14.5,denebola,11
60670 data168.6,15.4,leo2,11
60680 data168.5,20.5,leo3,11
60690 data155.0,19.9,leo4,11
60700 data152.1,12.0,regulus,11
60710 data151.8,16.8,leo7,11
60720 data146.5,23.8,leo6,11
60730 data116.3,28.0,pollux,12
60740 data113.7,31.9,kastor,12
60750 data101.3,12.9,gem3,12
60760 data101.0,25.1,gem4,12
60770 data 99.4,16.4,gem5,12
60780 data 95.7,22.5,gem6,12
60790 data111.0,-29.3,cma1,13
60800 data107.1,-26.4,cma2,13
60810 data104.7,-29.0,cma3,13
60820 data101.3,-16.7,sirius,13
60830 data 98.2,-18.0,cma5,13
60840 data90.0,37.2,aur1,14
60850 data89.9,45.0,aur2,14
60860 data79.2,46.0,capella,14
60870 data75.5,43.8,aur4,14
60880 data74.3,33.2,aur5,14
60890 data88.8, 7.4,beteigeuze,15
60900 data86.9,-9.1,ori2,15
60910 data85.2,-2.0,ori3,15
60920 data84.1,-1.2,ori4,15
60930 data83.0,-0.3,ori5,15
60940 data81.3, 6.4,bellatrix,15
60950 data78.6,-8.2,rigel,15
60960 data59.5,40.0,per1,16
60970 data58.5,31.9,per2,16
60980 data55.7,47.8,per3,16
60990 data51.1,49.9,algenib,16
61000 data47.0,41.0,algol,16
61010 data46.2,53.5,per6,16
61020 data31.0,42.0,alamak,17
61030 data17.4,35.6,mirach,17
61040 data 9.8,31.9,and3,17
```


Listing zu Sternenhimmel. (Fortsetzung)

```
61050 data 2.1,29.1,sirrah,17
61060 data354.8,77.6,cep1,18
61070 data342.5,66.0,cep2,18
61080 data322.2,70.6,alfrik,18
61090 data319.6,62.6,alderamin,18
61100 data269.2,51.5,dra1,19
61110 data262.6,52.3,dra2,19
61120 data231.2,59.0,dra3,19
61130 data246.0,61.5,dra4,19
61140 data257.2,65.7,dra5,19
61150 data288.1,67.7,dra6,19
61160 data31.8,23.5,hamal,20
61170 data28.7,20.8,ari2,20
61180 data84.4,21.2,tau1,21
61190 data81.6,28.6,elnath,21
61200 data69.0,16.5,aldebaran,21
61210 data58.8,12.5,tau4,21
61220 data56.9,24.1,plejaden,21
61230 data114.8,5.2,prokyon,22
61240 data111.8,8.3,cmi2,22
62000 rem ** daten sternbilder **
62010 dataursa minor,kleiner wagen,6,1
62020 dataursa maior,grosser wagen,6,8
62030 datacassiopeia,kassiopeia,4,15
62040 datapegasus,pegasus,2,20
62050 datacygnus,schwan,4,23
62060 dataaquilla,adler,5,28
62070 datalyra,leier,4,34
62080 datascorpius,skorpion,9,39
62090 databootes,bootes,5,49
62100 datavirgio,jungfrau,6,55
62110 datacrux,kreuz sueden,3,62
62120 dataleo,loewe,6,66
62130 datagemini,zwillinge,5,73
62140 datacanis maior,grosser hund,4,79
62150 dataauriga,fuhrmann,4,84
62160 dataorion,orion,6,89
62170 dataperseus,perseus,5,96
62180 dataandromeda,andromeda,3,102
62190 datacepheus,kepheus,3,106
62200 datadraco,drache,5,110
62210 dataaries,widder,1,116
62220 datataurus,stier,4,118
62230 datacanis minor,kleiner hund,1,123
62500 rem ** daten planeten **
62510 datamerkur,4.0923, 31.19, 76.987
62520 data23.00,.2056, 47.826,7.004, 0.3
871
62530 datavenus ,1.6021, 80.85,131.149
62540 data 0.76,.0068, 76.410,3.394, 0.7
233
62550 datamars  ,0.5240,144.14,335.507
62560 data11.00,.0934, 49.326,1.850, 1.5
237
62570 datajupiter,.0831,316.19, 13.839
62580 data 5.30,.0485,100.146,1.305, 5.2
028
62590 datasaturn,0.0335,158.36, 92.460
62600 data 5.50,.0557,113.511,2.486, 9.5
810
62610 datauranus,0.0117, 98.38,170.173
62620 data 5.70,.0472, 73.847,0.773,19.1
823
62700 rem ** e n d e **

ready.
```

Listing zu Sternenhimmel. Geben Sie das Listing nur dann ein, wenn Sie vorher Simons Basic geladen haben. (Schluß)



KREUZWORTRÄTSEL SELBER MACHEN

Kreuzworträtsel sind bei jung und alt beliebt. Das beweisen die vielen Rätselzeitschriften. Aber ein Kreuzworträtselprogramm zu schreiben ist etwas ganz anderes, als ein Rätsel zu lösen.

Das Programm läßt kaum Wünsche offen. Der Bildschirmaufbau ist genauso gut gelöst wie die Druckerausgabe. Auch den Bedienungskomfort kann man sich kaum besser vorstellen.

Ein Kreuzworträtselprogramm zu schreiben ist eine knifflige Aufgabe. Nicht nur die für den Betrachter eines Rätsels selbstverständlichen Regeln, wie das Kreuzen von Worten senkrecht zueinander und die dichte Vernetzung von Wörtern, auch die Geschwindigkeit und der Komfort des Programmes spielen eine große Rolle. Damit blieb in diesem Falle nur der Einsatz von Maschinensprache übrig. Das Kreuzworträtselprogramm besteht aus zwei Teilen. Einem Rahmenprogramm in Basic, sowie einer Maschinenroutine, die im Bereich ab $C000/49152 geladen wird und die die Schwerarbeit leistet (zirka 30 Wortvergleiche/Sekunde).

Um das Rätselprogramm nicht unnötig mit der Eingabe des Wortschatzes zu belasten, werden die Wörter vorher mit einem Editor eingegeben und bis zu 255 in einem File auf Diskette abgelegt. Das eigentliche Rätselprogramm liest diese Files dann ein und verwendet sie im Rätselfeld. Trotzdem ist die Eingabe von Hand möglich. Denn der große Wortschatz eines Menschen kann niemals auf Diskette gespeichert werden.

Die Programme

Bevor also die Erzeugung eines Kreuzworträtsels starten kann, muß ein Wortschatz mit dem Lexikon-Editor eingegeben werden (Listing 2). Dieses Programm hat vier Aufgaben:

a) Directory lesen. Dabei werden die bereits auf Diskette vorhandenen Wortschatzdateien angezeigt.

b) Wortdatei anlegen. Mit diesem Programmteil können Sie bis zu 255 Wörter und die zugehörigen Fragestellungen eingeben und diese dann auf Diskette als Wortdatei ablegen. Eine spezielle Eingaberoutine (Zeile 1 000 — 1 099) sorgt dafür, daß nur erlaubte Zeichen eingegeben werden. Es ist zum Beispiel unsinnig, ein Kreuzwort einzugeben, das aus mehreren Teilen besteht, wie etwa »Level II Basic«, oder eine Mischung aus Zahlen und Buchstaben, wie »C 64«.

So können Sie sich also im Laufe der Zeit eine ganze Bibliothek von Wortschätzen aufbauen, die beliebig während der Rätselerzeugung einzusetzen sind.

Eine Besonderheit ist die Verkettung von Dateien. Sie können Nachfolgedateien benennen, die im Rätselprogramm automatisch nachgeladen werden, sobald der Vorgängerwortschatz erschöpft ist. Beispiel: Sie wollen ein Rätsel zum Thema »Computer« basteln. Geben Sie jetzt mit Hilfe des Lexikon-Editors den ersten Teil des geplanten Wortschatzes ein und benennen ihn mit »Computer.1«. Der Folgewortschatz soll »Computer.2« sein. Nachdem Sie den 1. Teil abgespeichert haben, geben Sie den 2. Teil ein und benennen den Nachfolger mit »Computer.3« und so weiter. Wollen Sie diese Kette beenden, wird einfach ein »q« (Quit) für den Nachfolger angegeben. Das Rätselprogramm bricht dann später an dieser Stelle mit dem automatischen Nachladen ab.

Bei dieser Methode hängt die Qualität des Rätsels entscheidend von der Staffelung der Dateien ab, das heißt im ersten File sollten lange Wörter stehen und mit jedem Folgefile die durchschnittliche Wortlänge abnehmen. Die Erstellung des Rätsels gerät dadurch wesentlich flüssiger, denn mit zunehmender Dichte des Wortfeldes müssen die Wörter natürlich immer kürzer werden, damit sie noch einzubauen sind.

c) Wortdatei ändern. Mit Sicherheit tauchen irgendwann falsch eingegebene Kreuzwörter auf, so daß eine Datei verbessert werden muß. Unter diesem Programmpunkt kann daher die entsprechende Wortdatei geladen und editiert werden.

d) Abbrechen. Mit Verlassen des Editors kann das Rätselprogramm geladen und gestartet werden, so daß ein fließender Übergang von der Worteingabe zur Rätselerstellung möglich ist.

Nun zum Kreuzworträtselprogramm. Wie schon gesagt, besteht dieses Programm aus zwei Teilen, einer Maschinenroutine und dem Basic-Rahmenprogramm. Dieses Rahmenprogramm (Listing 1) will ich zuerst beschreiben.

Nach dem Start beginnt zunächst eine Initialisierungsphase (Zeilen 27—39). In einem Unterprogramm (ab Zeile 970) werden die Maschinenroutine für das Kreuzworträtsel und Treibersoftware für den Drucker am User-Port geladen. Wenn Sie einen Drucker am IEC-Bus betreiben, ist dieser Programmteil (Zeilen 976+979) natürlich nicht nötig und zu entfernen.

Im folgenden Unterprogramm zur Dimensionierung der Variablen und Definition einiger Konstanten (Zeilen 915—969) erfolgt der erste Sprung in die Maschinenroutine (Zeile 937), um sie zu initialisieren.

Nach dem Bildaufbau (Unterprogramm ab Zeile 600), ist auf dem Monitor ein Kreuzwortfeld von 20x20 Zeichen zu sehen. Das Programm fragt jetzt nach dem Startwort, das als erstes Wort im Feld eingetragen werden muß, um der Maschinenroutine einen Kristallisationspunkt zu bieten (Zeilen 34-35). Geben Sie ein möglichst langes Wort ein, das Sie zum Beispiel unbedingt im Rätsel verarbeiten möchten. Bei dieser Eingabe findet, wie schon beim Editor, ein Zeichencheck statt, der es unmöglich machen soll, unerlaubte Zeichen im Rätselfeld unterzubringen. Danach müssen Sie sich zwischen einer waagerechten oder senkrechten Eintragung entscheiden, und ein Cursor taucht im Wortfeld auf. Fahren Sie wie gewohnt mit den Cursortasten die gewünschte Wortposition an und drücken die RETURN-Taste. Da es das erste Wort ist und genügend Platz im Wortfeld herrscht, wird Ihr Wort ohne Protest sofort eingetragen. Jetzt noch schnell die zugehörige Fragestellung eingetippt, und das erste Wort ist korrekt eingetragen.

Sie können dieses Spiel beliebig fortsetzen und theoretisch das ganze Rätsel auf diese Weise per Hand erstellen. Alle Eingaben von Hand sind frei von dem Zwang, ein Wort mit einem anderen kreuzen zu müssen. Sie können Ihre Wörter also beliebig positionieren, sollte ein Wort allerdings nicht passen, wird es zurückgewiesen und der Handeingabe-Modus abgebrochen.

Regulär verlassen Sie die Handeingabe, indem Sie anstelle eines neuen Wortes ein »q« eintippen. Später können Sie die automatische Rätselerzeugung jederzeit unterbrechen und mit F7 wieder in den Handmodus zurückkehren.

Im Wortfeld stehen nun ein oder mehrere Wörter, die als Startpunkte für andere Wörter dienen. Senkrechte Eintragungen sind weiß und waagerechte gelb gefärbt.

Bleibt nur die Angabe, welcher auf Diskette gespeicherte Wortschatz als erstes geladen werden soll (Zeile 37). Dabei können Sie sich mit »$« auch das Inhaltsverzeichnis der Dis-

kette ansehen, für den Fall, daß Ihnen der Name eines Wortschatzes entfallen ist.

Nach erfolgreichem Laden des ersten Wortschatzes beginnt nun die automatische Rätselerzeugung (Hauptschleife Zeile 40-99). Zunächst überraschend schnell füllt sich das Wortfeld mit zufällig plazierten und gekreuzten Wörtern. Dieser Vorgang wird von der Maschinenroutine gesteuert. Auf die genaue Arbeitsweise dieses Programmteiles gehe ich noch gesondert ein. Links oben auf dem Bildschirm erscheint die Anzahl der Wörter, die bereits eingetragen sind (maximal 255 sind möglich) und daneben die momentane prozentuale Größe des noch zur Verfügung stehenden Wortschatzes im Speicher.

Bei jeder Eintragung wird die Liste der Kreuzwörter gekürzt und das benutzte Wort aus der Liste gestrichen. Dies geschieht durch Vertauschen der Stringvektoren des zu streichenden Wortes und des letzten Wortes im Array (Zeile 300—399). Diese Methode wurde gewählt, um die Bildung von neuen Strings im Speicher zu vermeiden und der schrecklich langsamen Garbage-Collection aus dem Wege zu gehen.

Während der Rätselerzeugung können Sie sich in Ruhe überlegen, ob Sie ein automatisches Nachladen von Wortschätzen gestatten wollen oder nicht. In der rechten Bildhälfte ist inzwischen die Belegung der Funktionstasten zu sehen, und mit der F6-Taste schalten Sie die Autolader-Option ein oder aus. Erscheint »Aus« im F6-Tastenfeld, ist das Nachladen gesperrt.

Die Maschinenroutine sucht derweil ständig nach passenden Stellen im Wortfeld. Dieser Vorgang kann im Prinzip endlos sein, da irgendwann natürlich kein geeignetes Wort mehr zu finden ist. Aus diesem Grund hat das Programm eine »Geduld-Schwelle«, eine Anzahl von Suchversuchen, innerhalb derer ein passendes Wort gefunden werden muß. Ist die Versuchszahl ohne Erfolg abgelaufen, geht das Programm davon aus, daß der Wortschatz nicht mehr ausreicht. Diese »Geduld-Schwelle« können Sie in Zeile 964 selbst bestimmen.

Bevor allerdings in der obersten Bildzeile die Meldung »Wortschatz ungenügend« erscheint (Zeile 64/Unterprogramm 110-158), hebt das Programm noch die »Blausperre« auf (Zeile 66). Dieses Flag hat dem Maschinenprogramm bisher mitgeteilt, daß nur dann eine Eintragung erlaubt ist, wenn dabei auch ein blaues, unbesetztes Feld abgedeckt wird. Eine Maßnahme, um das Rätsel möglichst dicht zu packen. Ab sofort ist also auch das Einpassen eines Wortes nur auf besetzten Feldern möglich. Läßt sich auch jetzt kein Wort mehr finden, erscheint endgültig der Hinweis auf mangelnde Wortauswahl. Wenn Sie das »Autoladen« zugelassen haben, wird nun der Folgewortschatz, falls vorhanden, gelesen und das Spiel beginnt von neuem. Selbstverständlich ist auch das Laden von Wortdateien vor Ablauf der »Geduld-Schwelle« machbar. Dazu dient die F1-Taste. Aber Vorsicht, auf die Gefahr, daß Sie einen bereits verbrauchten Wortschatz noch mal laden, müssen Sie schon selbst achten.

Es dauert gar nicht so lange, dann ist das Wortfeld so dicht gepackt, daß der Maschinenroutine keine Eintragung mehr gelingt. Jetzt sind Sie gefordert, und mit der F7-Taste wählen Sie die Handeingabe an. Genau wie bei der Eintragung der ersten Startwörter können Sie Ihre »Lückenfüller« positionieren und die Fragestellung dazu eingeben.

Zufrieden mit Ihrer Arbeit (hoffentlich!) bleibt nur noch der Ausdruck des Rätsels. Mit F4 wird er gestartet. Das Unterprogramm für die Druckausgabe nimmt im Programm den weitaus größten Platz ein (Zeile 740-914). Es ist für einen Epson RX-80 ausgelegt, der mit Hilfe spezieller Treibersoftware auch CBM-Sonderzeichen druckt. Dazu muß eine unübliche Geräteadresse (6) angegeben werden (Zeile 761 und 900). Sollten Sie also einen CBM-grafikfähigen Drucker am IEC-Bus betreiben, tauschen Sie diese Adresse gegen die gewohnte »4« (im Listing schon geändert).

Auch die Druckerinitialisierung ist von Drucker zu Drucker verschieden (Zeilen 754—758). Achten Sie darauf, daß Ihr Drucker hier folgende Einstellung erhält:
— Zeilenabstand = 0
— CBM-Grafikmodus
— Startposition des Druckkopfes = lr (Linker Rand, kann in Zeile 946 geändert werden).

Für den Ausdruck der Fragestellung wird der RX-80 im Engschriftmodus versetzt, um Platz zu sparen (Zeile 824—827). Auf diesen Effekt können Sie natürlich verzichten, müssen dann aber die Zeilenlänge (cz, in Zeile 948) ändern, da der Ausdruck vom Programm mit Randausgleich versehen wird (Zeile 847 — 889).

Nun wie versprochen zum Maschinenprogramm, kurz genannt »Such—obj« (Listing 3). Diese Routine liegt im Bereich $C000/49152, wo sie gut gegen Überschreiben durch Basic geschützt ist. Der Einsprung erfolgt über den USR-Vektor (definiert in Zeile 931 bis 932), das heißt es findet eine Parameterübergabe zwischen Basic und Maschinenroutine statt. Basic übergibt in »dl« die Anzahl der Versuche, die die Routine durchlaufen sollen (Zeile 44). Mit der Rückkehr aus der Routine wird der Variablen »u« ein Wert zwischen 1 und 4 zugewiesen. Aus dem Wert von »u« kann also auf die Ursache für den Abbruch der Routine geschlossen werden:
— u = 1; eine Taste ist betätigt worden.
— u = 2; die Suche nach einem passenden Kreuzwort war vergeblich.
— u = 3; ein Wort wurde gefunden und in das Wortfeld eingetragen.
— u = 4; die maximale Anzahl (255) von eingetragenen Kreuzwörtern ist erreicht; keine weitere Eintragung möglich.

Das Rahmenprogramm kann jetzt entsprechend reagieren und zum Beispiel im Falle u = 3 das benutzte Wort aus der Wortliste streichen. Im Falle einer Eintragung (von Hand oder automatisch) wird das Wort nicht nur in den Bildspeicher eingesetzt, es werden noch weitere Informationen abgelegt:

a) Eintragung im »Wortbeginn/-ende«-Speicher (50176 bis 50575); hier wird vermerkt, ob ein Rätselfeld den Start- oder Endpunkt eines Kreuzwortes repräsentiert. Der Speicher ist, wie die folgenden auch, in 20 Zeilen mit je 20 Positionen (Speicherstellen) aufgeteilt. Das linke Halbbyte (4 Bits) einer Speicherstelle trägt die Informationen über senkrechte, das rechte Halbbyte über waagerechte Start-/Endpositionen. Das 1. Bit im Halbbyte wird für Endpunkte gesetzt, das 3. Bit für Startpunkte.
Diese Informationen werden später bei der Ausgabe des Rätsels auf dem Drucker benötigt, um an den richtigen Stellen die Nummer der zugehörigen Fragestellung eintragen zu können.

b) Eintragungen im »Senkrecht«-Speicher (50576 bis 50975); in dieser Speichermatrix werden die Ordnungsnummern der Fragestellungen für senkrechte Rätselwörter abgelegt. Bei der Druckausgabe wird diese Matrix abgefragt (Zeile 773 bis 777), um die Nummer der Fragestellung im entsprechenden Startfeld einzusetzen.

c) Eintragung im »Waagerecht«-Speicher (50976 bis 51375), wie unter b) Zeile 796 bis 800).

Um die Speicherinhalte vor dem Start des Rätselprogramms zu löschen, springt man die Routine mit »sys 50016« an (Zeile 937).

Nun zur Arbeitsweise der Routine während der automatischen Rätselerzeugung. Zunächst muß das Maschinenprogramm erst einmal wissen, ob es wegen einer Handeintragung angesprungen wurde oder zur automatischen Wortsuche. Dazu liest die Routine die Speicherstelle 26 aus. Ist das Ergebnis Null, wird in den Automodus verzweigt. Andernfalls wird in 26 die Länge des von Hand eingetragenen Wortes übergeben (siehe Zeile 662 bis 663). Die Routine braucht dann nur den Paß des Wortes zu überprüfen und bei korrekter Eintragung ins Basic zurückzukehren. Paßt das Wort allerdings nicht, ver-

fällt die Routine in den Automodus. Das Maschinenprogramm durchläuft im Automodus eine Schleife, die zunächst durch Auslesen des Rauschgenerators im Soundchip ein Wort aus dem Wortschatz per Zufall bestimmt. Dann erfolgt auf die gleiche Weise die Auswahl eines Startpunktes im Bildspeicher. Ist dieser Punkt nicht geeignet für eine Eintragung, wird Zeile für Zeile des Wortfeldes nach einer Alternative gesucht. Im Falle eines Treffers startet der Wortvergleich. Das gewählte Wort wird mit dem Inhalt des Bildspeichers auf Übereinstimmung geprüft. Paßt es nicht, kommt das nächste Wort aus dem Wortschatz-Array an die Reihe. Bei Erfolg kehrt »Such« ins Basic zurück, wenn nicht, beginnt die Schleife von neuem.

Damit der Zugriff auf das Array klappt, muß es nur als erstes im Basic-Programm definiert worden sein (Zeile 925). »Primitiv« werden Sie vielleicht anmerken. Richtig, aber Computer sind nun mal (sehr) schnelle Idioten.

(Gert Büttgenbach/og)

```
15 PRINT"{CLR}";                                   <209>
16 PRINT"R******************************S"         <081>
17 PRINT"-KREUZWORTRAETSEL VERSION 1.A-"           <221>
18 PRINT"-{28SPACE}-                               <033>
19 PRINT"-(C) BUETTGENBACH{2SPACE}HH,30.5.
   84-                                             <173>
20 PRINT"Z******************************X         <066>
22 :                                               <254>
23 :                                               <255>
24 REM" ******** HAUPTPROGRAMM ********            <066>
25 REM" ******************************             <011>
26 :                                               <002>
27 REM" ******* VORBEREITUNGEN ********            <004>
28 CLR                                             <142>
29 GOSUB 970 :REM" ROUTINEN LADEN                  <100>
30 GOSUB 915 :REM" VORBEREITUNGEN                  <049>
32 GOSUB 600 :REM" BILD AUFBAUEN                   <187>
33 :                                               <009>
34 T$="STARTWORT" :REM" 1.WORT                     <251>
35 NR=1:GOSUB 654  :REM" EINTRAGEN                 <253>
36 :                                               <012>
37 GOSUB 700 :REM" WORTSCHATZ EINLESEN             <088>
38 POKE 53280,0:POKE 53281,0                       <166>
39 :                                               <015>
40 REM" ******** HAUPTSCHLEIFE ********            <009>
42 :                                               <018>
44 POKE 142,BS:POKE 26,0:U=USR(DL)                 <127>
46 :                                               <022>
47 ON U GOTO 50,64,70,80                           <052>
48 :                                               <024>
49 REM" ***** TASTE WAR GEDRUECKT *****            <197>
50 GET F$                                          <044>
51 IF F$="{F7}"THEN GOSUB 650:GOTO 59              <161>
52 IF F$="{F4}"THEN GOSUB 740:GOTO 59              <155>
53 IF F$="{F1}"THEN GOSUB 700:GOTO 60              <200>
54 IF F$="{F6}"THEN GOSUB 450:GOTO 60              <004>
55 IF F$="{F8}"THEN GOSUB 092:GOTO 59              <247>
56 IF F$="{F2}"THEN GOSUB 087:GOTO 59              <006>
57 IF FG>0 THEN 44                                 <190>
58 GOTO 50                                         <012>
59 PRINT"{HOME}";LL$;LL$                           <069>
60 POKE 53280,0:POKE 53281,0:GOSUB 337             <050>
61 IF FG>0 THEN 44                                 <194>
62 GOTO 50                                         <016>
63 :                                               <039>
64 REM" **** SUCHE WAR VERGEBLICH *****            <044>
65 IF BS=0 THEN GOSUB 110                          <031>
66 BS=0:GOTO 44                                    <004>
69 :                                               <045>
70 REM" *** EIN WORT WURDE GEFUNDEN ***            <103>
71 W=PEEK(25)                                      <185>
72 FS$(NR)=STR$((PEEK(140)=0))+FW$(W)              <079>
73 GOSUB 300:NR=NR+1:IF FG>0 THEN 44               <163>
75 :                                               <051>
76 PRINT"{HOME}";LL$;"{HOME}";TAB(17);             <237>
77 PRINT"{CYAN,RVSON}WORTSCHATZ ERSCHOEPFT
   !"                                              <163>
78 GOSUB 337:GOTO 50                               <029>
79 :                                               <055>
80 REM" *** MAX. ORDN.ZAHL ERREICHT ***            <044>
81 PRINT"{HOME}";LL$;"{HOME,CYAN,RVSON}RAE
   TSEL FERTIG:{RVOFF}";                           <012>
82 PRINT"{2SPACE}AUSDRUCKEN MIT{SPACE,RVSO
   N}RETURN{RVOFF}"                                <229>
83 WAIT 203,64:WAIT 203,63                         <002>
84 PRINT"{HOME}";LL$;LL$:GOSUB 740                 <017>
85 GOTO 95                                         <151>
86 :                                               <062>
87 REM" ****** RAETSEL LOESCHEN *******            <106>
88 F$="RAETSEL LOESCHEN":GOSUB 4000                <240>
89 IF F$<>"J"THEN RETURN                           <142>
90 RUN 30                                          <032>
91 :                                               <067>
92 REM" ***** PROGRAMM ABBRECHEN ******            <127>
93 F$="ABBRECHEN":GOSUB 4000                       <043>
94 IF F$<>"J"THEN RETURN                           <147>
95 GOSUB 500:REM" PROGRAMM-ENDE                    <239>
96 :                                               <072>
97 END                                             <099>
98 :                                               <074>
99 :                                               <075>
100 REM" ******* UNTERPROGRAMME *******            <186>
102 REM" ******************************            <176>
104 :                                              <080>
106 :                                              <082>
110 REM" AUTOM. NACHLADEN                          <026>
112 REM" ******************************            <174>
114 :                                              <090>
115 PRINT"{HOME,CYAN}";LL$;"{HOME}";TAB(18
    );                                             <010>
116 PRINT"{RVSON}WORTSCHATZ UNGENUEGEND"           <090>
117 GOSUB 337                                      <093>
118 IF RIGHT$(NF$,1)="Q"THEN 154                   <185>
120 :                                              <096>
122 DL=INT(DM/3)                                   <183>
124 :                                              <100>
125 IF AV=0 THEN AV=1:GOTO 154                     <125>
126 IF LS=0 THEN 154                               <227>
127 :                                              <103>
128 REM" *** FOLGE-WORTSCHATZ LADEN ***            <115>
130 POKE 53280,2:POKE 53281,2                      <014>
132 PRINT"{HOME}";LL$;"{HOME,CYAN}WORTSCHA
    TZ ";                                          <156>
133 PRINT"{RVSON}";NF$;"{RVOFF,SPACE}WIRD
    GELADEN"                                       <238>
136 F$=NF$:GOSUB 713                               <070>
137 POKE 53280,0:POKE 53281,0                      <009>
138 :                                              <114>
154 RETURN                                         <212>
156 :                                              <132>
158 :                                              <134>
300 REM" WORTLISTE KUERZEN                         <167>
302 REM" ******************************            <110>
304 :                                              <026>
307 SA=PEEK(47)+PEEK(48)*256+7                     <075>
308 :                                              <030>
315 GOSUB 350 :REM" WOERTER KUERZEN                <150>
316 :                                              <038>
320 SA=SA+772                                      <064>
322 GOSUB 350 :REM" FRAGEN KUERZEN                 <045>
324 :                                              <046>
334 FG=FG-1:POKE 2,FG:REM" FELD KUERZEN            <134>
336 :                                              <058>
337 PRINT"{HOME,WHITE}";NR;INT(FG/OG*100);
    "{LEFT,SPACE}%{2SPACE}"                        <070>
338 :                                              <060>
341 RETURN                                         <145>
342 :                                              <064>
344 :                                              <066>
349 REM" WORTVEKTOREN VERTAUSCHEN                  <056>
350 X=SA+FG*3:Y=SA+W*3                             <191>
354 POKE Y,PEEK(X)                                 <005>
356 POKE Y+1,PEEK(X+1)                             <103>
358 POKE Y+2,PEEK(X+2)                             <145>
360 :                                              <082>
397 RETURN                                         <201>
398 :                                              <120>
399 :                                              <171>
450 REM" AUTOLADEN EIN/AUS                         <059>
452 REM" ******************************            <004>
454 :                                              <176>
456 IF LS=0 THEN LS=1:GOTO 461                     <054>
458 IF LS=1 THEN LS=0                              <067>
460 :                                              <182>
461 POKE 214,13:POKE 211,30:SYS 58732              <130>
462 :                                              <184>
463 PRINT"{RED}";                                  <027>
464 IF LS=1 THEN PRINT"{RVSON}EIN{RVOFF}";         <172>
466 IF LS=0 THEN PRINT"AUS";                       <163>
472 :                                              <194>
497 RETURN                                         <045>
498 :                                              <220>
499 :                                              <221>
```

Listing 1. »Raetsel«. Beachten Sie die Eingabehinweise auf Seite 16.

Listing 1. »Raetsel«.
Beachten Sie die Eingabehinweise auf Seite 16.

```
500 REM" PROGRAMM-ENDE                      <241>
502 REM" *******************************    <054>
504 :                                       <226>
505 PRINT"{CLR,LIG.BLUE}";CHR$(9);          <026>
506 POKE 53280,14:REM" RAHMEN HELLBLAU      <152>
508 POKE 53281,06:REM" GRUND BLAU           <228>
510 :                                       <232>
512 POKE 792,71:REM" RESTORE UND            <119>
514 POKE 788,49:REM" STOP MOEGLICH          <157>
594 :                                       <062>
596 RETURN                                  <146>
598 :                                       <066>
599 :                                       <067>
600 REM" BILD AUFBAUEN                      <121>
602 REM" *******************************    <156>
604 :                                       <072>
606 POKE 53265,11 :REM" BILD AUS            <084>
607 :                                       <075>
608 REM" ****** KREUZWORTFELD *********     <104>
609 PRINT"{CLR,3DOWN}";CHR$(14);CHR$(8);    <110>
610 PRINT"{SPACE,CYAN}R*******************
    *S "                                    <132>
612 FOR N=1 TO 20                           <163>
614 PRINT"{SPACE,CYAN}-{RVSON,LIG.BLUE,20S
    PACE,RVOFF,CYAN}-"                      <191>
616 NEXT                                    <118>
618 PRINT"{SPACE,CYAN}Z*******************
    *X ";                                   <026>
619 :                                       <087>
620 REM" ***** TASTENBELEGUNG *********     <148>
621 B$="{28RIGHT}"                          <178>
622 PRINT"{HOME,3DOWN,RED}";                <084>
623 A$="WOERTER{3SPACE}EINLESEN"            <195>
624 F=1:GOSUB 639                           <159>
625 A$="RAETSEL{3SPACE}DRUCKEN "            <206>
626 F=4:GOSUB 639                           <209>
627 A$="AUTOLADER AUS{5SPACE}"              <054>
628 F=6:GOSUB 639                           <243>
629 A$="VON HAND{2SPACE}EINGEBEN"           <049>
630 F=7:GOSUB 639                           <005>
631 PRINT B$;"{DOWN,RVSON}F 2{RVOFF,DOWN,2
    LEFT}LOESCHEN"                          <071>
632 PRINT B$;"{DOWN,RVSON}F 8{RVOFF,DOWN,2
    LEFT}ABBRECHEN";                        <080>
633 :                                       <101>
634 POKE 53265,27:REM" BILD AN              <029>
635 :                                       <103>
636 RETURN                                  <186>
637 :                                       <105>
639 PRINT B$;"{RVSON}F";F;"{RVOFF}******S   <003>
640 REM    PRINTB$;"-{9SPACE}-              <110>
641 PRINT B$;"-";LEFT$(A$,9);"-             <108>
642 PRINT B$;"-";RIGHT$(A$,9);"-"           <147>
643 PRINT B$;"Z**********X                  <063>
645 :                                       <113>
646 RETURN                                  <196>
648 :                                       <116>
649 :                                       <117>
650 REM" KREUZWORT V. HAND EINGEBEN         <088>
651 REM" *******************************    <205>
652 :                                       <120>
653 T$="KREUZWORT (ABBR=Q)"                 <062>
654 POKE 53280,2:POKE 53281,2               <030>
655 PRINT"{HOME}";LL$;LL$                   <157>
656 L=20:M=0:GOSUB 1000:M$=F$               <059>
657 IF M$="Q"AND NR>1 THEN 685              <040>
658 L=LEN(M$):IF L<2 THEN 656               <085>
659 :                                       <127>
660 T$=M$:GOSUB 2000 :REM" POS. EINGEBEN    <035>
661 :                                       <129>
662 POKE 26,LEN(M$):M$=M$:REM" WORT         <044>
663 WAIT 203,64:U=USR(0) :REM" EINTRAGEN    <179>
664 IF U<>3 THEN 685                        <254>
665 :                                       <133>
666 T$="FRAGESTELLUNG" :REM" FRAGE          <118>
667 L=36:M=1:GOSUB 1000 :REM" EINGEBEN      <023>
669 :                                       <137>
680 X$=STR$(PEEK(140)=0)   :REM" FRAGE      <146>
681 FS$(NR)=X$+F$:NR=NR+1 :REM" MERKEN      <084>
682 AV=0:DL=DM                              <114>
683 GOTO 650                                <223>
684 :                                       <152>
685 RETURN                                  <235>
698 :                                       <166>
699 :                                       <167>
700 REM" KREUZWOERTER EINLESEN              <072>
701 REM" *******************************    <255>
702 :                                       <170>
703 POKE 53280,2:POKE 53281,2               <079>
704 PRINT"{HOME}";LL$;LL$;                  <156>
705 T$="WORTSCHATZ  (ABBR=Q)"               <045>
706 L=14:M=1:GOSUB 1000                     <254>
707 IF F$="Q"THEN PRINT"{HOME}";LL$:GOTO 7
    33                                      <173>
708 IF F$="$"THEN 710                       <051>
709 F$="±"+CHR$(20)+F$:GOTO 712             <168>
710 GOSUB 3000:IF F$=""THEN 704             <137>
711 :                                       <179>
712 PRINT"{HOME}";LL$;LL$                   <214>
713 OPEN 15,8,15                            <107>
714 OPEN 1,8,2,"0:"+F$+",S,R"               <169>
715 INPUT#15,EN,EN$                         <202>
716 IF EN<>0 THEN 728                       <164>
717 :                                       <185>
718 INPUT#1,FG :REM" FELDGROESSE GLEICH     <226>
719 POKE 2,FG    :REM" ANZAHL WOERTER       <166>
720 FOR N=0 TO FG :REM"LESE WOERTER/FRAGEN  <175>
721 :INPUT#1,KW$(N),FW$(N)                  <197>
722 NEXT                                    <224>
723 :                                       <191>
724 INPUT#1,NF$:REM" NAECHSTER FILE         <136>
725 INPUT#15,EN,EN$                         <212>
726 AV=0:DL=DM:BS=1:OG=FG                   <150>
727 :                                       <195>
728 PRINT"{HOME}";LEFT$(LL$,40-LEN(EN$));   <029>
729 PRINT"{RVSON}";EN$                      <191>
730 :                                       <198>
731 CLOSE 1:CLOSE 15                        <135>
732 :                                       <200>
733 RETURN                                  <027>
734 :                                       <202>
735 :                                       <203>
740 REM" RAETSEL DRUCKEN                    <084>
741 REM" *******************************    <039>
742 :                                       <210>
743 POKE 53280,2:POKE 53281,2               <119>
744 PRINT"{HOME}";LL$;LL$                   <246>
745 PRINT"{HOME,CYAN,RVSON}RAETSEL WIRD AU
    SGEDRUCKT{RVOFF}"                       <028>
746 :                                       <214>
747 GOSUB 761:REM" WORTFELD AUSDRUCKEN      <038>
748 GOSUB 824:REM" FRAGEN AUSDRUCKEN        <082>
749 GOSUB 900:REM" LOESUNG AUSDRUCKEN       <003>
750 :                                       <218>
751 RETURN                                  <045>
752 :                                       <220>
753 :                                       <221>
754 REM" ** INIT EPSON RX-80 DRUCKER **     <011>
755 OPEN 1,4:PRINT#1,CHR$(27);"@";          <252>
756 PRINT#1,CHR$(27);"3";CHR$(24);          <198>
757 REM PRINT#1,CHR$(27);"L";CHR$(LR);      <150>
758 PRINT#1: CLOSE 1:RETURN                 <147>
759 :                                       <227>
760 REM" **** WORTFELD AUSDRUCKEN *****     <252>
761 GOSUB 755:OPEN 1,4,10:CMD 1 :PRINT:PRI
    NT"{RVSON}R";                           <158>
762 FOR N=0 TO 19:PRINT"***";:NEXT          <170>
763 PRINT"S{RVOFF}"                         <058>
764 :                                       <232>
765 FOR ZE=0 TO 19                          <132>
766 :D=ZE*20:AD=S1+D:AS=S2+D:AW=S3+D        <221>
767 :AB=S4+ZE*40                            <126>
768 :                                       <238>
772 :PRINT" ";                              <242>
773 :FOR SP=0 TO 19                         <236>
774 :  C=PEEK(AS+SP)                        <068>
775 :  IF C=0 THEN PRINT"{3SPACE}";:GOTO 77
    7                                       <107>
776 :  PRINT MID$(STR$(C)+"{3SPACE}",2,3);  <194>
777 :NEXT SP                                <074>
779 :GOSUB 820                              <058>
780 :FOR N=1 TO 3                           <017>
781 :  PRINT"{RVSON}-{RVOFF}";              <131>
782 :  FOR SP=0 TO 19                       <245>
783 :    A=166:B=A:C=A                      <233>
784 :    IF PEEK(AB+SP)=160 THEN 790        <070>
785 :    B=32:C=B:IF SP=19 THEN 787         <191>
786 :    IF(PEEK(AD+SP)AND 1)THEN C=167     <120>
787 :    A=165:IF SP=0 THEN A=32            <061>
788 :    IF(N=1)AND PEEK(AS+SP)THEN A=32    <249>
789 :    IF(N=3)AND PEEK(AW+SP)THEN A=32    <123>
790 :    PRINT CHR$(A);CHR$(B);CHR$(C);     <048>
791 :  NEXT SP                              <088>
792 :  PRINT"{RVSON}-{RVOFF}";:IF N<3 THEN
    PRINT                                   <100>
793 :NEXT N                                 <255>
```

Listing 1. »Raetsel« (Fortsetzung)

```
794 :                                            <008>
795 :GOSUB 820:PRINT" ";                         <165>
796 :FOR SP=0 TO 19                              <003>
797 : C=PEEK(AW+SP)                              <095>
798 : IF C=0 THEN PRINT"{3SPACE}";:GOTO 80
    0                                            <057>
799 : PRINT MID$(STR$(C)+"{3SPACE}",2,3);        <217>
800 :NEXT                                        <129>
801 :                                            <015>
802 :IF ZE=19 THEN 810                           <241>
803 :GOSUB 820:PRINT" ";                         <173>
804 :FOR SP=0 TO 19                              <011>
805 : P=AD+SP:C=(PEEK(P)AND 16)                  <168>
806 : IF C THEN PRINT"PPP";:GOTO 808             <004>
807 : PRINT"@@@";                                <170>
808 :NEXT SP                                     <105>
809 :                                            <023>
810 PRINT:NEXT ZE                                <043>
811 :                                            <025>
812 PRINT"{RVSON}Z";                             <164>
813 :FOR N=0 TO 19:PRINT"***";:NEXT              <116>
814 PRINT"X{RVOFF}";:PRINT#1:CLOSE 1             <045>
815 :                                            <029>
816 RETURN                                       <112>
817 :                                            <031>
818 :                                            <032>
820 PRINT CHR$(141);:RETURN                      <038>
821 REM  PRINTCHR$(141);LEFT$(LL$,LR);:RET
    URN :REM FUER RX80                           <200>
822 :                                            <036>
823 REM" ***** FRAGEN AUSDRUCKEN ******          <041>
824 PRINT CHR$(14):OPEN 1,4                      <012>
825 PRINT#1,CHR$(27);"0";                        <162>
826 PRINT#1,CHR$(27);CHR$(15);                   <127>
827 PRINT#1,CHR$(27);"L";CHR$(LR*1.7);           <042>
828 :                                            <042>
829 CMD 1:PRINT CHR$(14);"SENKRECHT:"            <018>
830 SW=0:GOSUB 838:PRINT                         <184>
831 :                                            <045>
832 PRINT CHR$(14);"WAAGERECHT:"                 <006>
833 SW=-1:GOSUB 838:PRINT#1:CLOSE 1              <002>
834 :                                            <048>
835 RETURN                                       <131>
836 :                                            <050>
837 :                                            <051>
838 MZ=0:ZL=0:F=0:NA=1                           <135>
839 FOR N=1 TO NR-1                              <211>
840 :IF VAL(FS$(N))=SW THEN GOSUB 847            <064>
841 NEXT N                                       <203>
842 M=0:MR=0:BZ=0:GOSUB 881:PRINT                <076>
843 :                                            <057>
844 RETURN                                       <140>
845 :                                            <059>
846 :                                            <060>
847 NL=LEN(STR$(N))+2                            <160>
848 L=ZL+NL+LEN(FS$(N))                          <059>
849 IF L<CZ THEN ZL=L:GOTO 861                   <063>
850 :                                            <064>
851 PA=1                                         <245>
852 FOR P=3 TO LEN(FS$(N))-2                     <215>
853 :IF MID$(FS$(N),P,1)<>" "THEN 855            <073>
854 :L=ZL+NL+P-2:IF L<CZ THEN PA=P               <248>
855 NEXT P                                       <233>
856 :                                            <070>
857 IF PA=1 THEN MZ=MZ-1:L=ZL                    <204>
858 IF PA>1 THEN L=ZL+NL+PA-2                    <225>
859 GOSUB 866:MZ=0                               <052>
860 :                                            <074>
861 MZ=MZ+1                                      <090>
862 :                                            <076>
863 RETURN                                       <159>
864 :                                            <078>
865 :                                            <079>
866 BZ=INT((CZ-L)/MZ):MR=CZ-(L+BZ*MZ)            <209>
867 M=-(MR>0):GOSUB 881                          <121>
868 :                                            <082>
869 NA=N+1:L=LEN(FS$(N)):IF PA>1 THEN 873        <146>
870 W=N:PRINT";":GOSUB 892:ZL=NL+L+3             <008>
871 PRINT MID$(FS$(N),3,L);:GOTO 878             <145>
872 :                                            <086>
873 PRINT"; ";LEFT$(LL$,BZ);                     <144>
874 W=N:GOSUB 892:ZL=L-PA+2                      <114>
875 PRINT MID$(FS$(N),3,PA-2)                    <069>
876 PRINT RIGHT$(FS$(N),L-PA);                   <171>
877 :                                            <091>
878 RETURN                                       <174>
879 :                                            <093>
880 :                                            <094>
881 FOR P=NA TO N-1                              <254>
882 :IF VAL(FS$(P))<>SW THEN 887                 <044>
883 :IF F THEN PRINT"; ";LEFT$(LL$,BZ+M);        <162>
884 :IF F THEN MR=MR+(MR>0):M=-(MR>0)            <245>
885 :F=1:W=P:GOSUB 892                           <251>
886 :PRINT RIGHT$(FS$(P),LEN(FS$(P))-2);         <134>
887 NEXT P                                       <009>
888 :                                            <102>
889 RETURN                                       <185>
890 :                                            <104>
891 :                                            <105>
892 NL=LEN(STR$(W))-1:PRINT"(";                  <252>
893 PRINT RIGHT$(STR$(W),NL);") ";               <207>
894 RETURN                                       <190>
895 :                                            <109>
899 REM" ***** LOESUNG AUSDRUCKEN *****          <060>
900 GOSUB 755:PRINT CHR$(14):OPEN 1,4,10:C
    MD 1                                         <182>
901 :                                            <115>
902 PRINT:PRINT:PRINT:PRINT"LOESUNG:"            <045>
903 PRINT"R********************S"                <088>
904 FOR N=S4 TO S4+760 STEP 40                   <053>
905 :PRINT"=";:FOR P=N TO N+19                  <091>
906 :PRINT CHR$(PEEK(P));                        <168>
907 :NEXT P:PRINT"="                             <113>
908 NEXT N                                       <014>
909 PRINT"Z********************X"                <075>
910 PRINT#1:CLOSE 1                              <130>
911 :                                            <125>
912 RETURN                                       <208>
913 :                                            <127>
914 :                                            <128>
915 REM" VORBEREITUNGEN                          <057>
916 REM" ******************************          <216>
917 :                                            <131>
919 POKE 53280,2:REM" RAHMEN GRAU                <003>
920 POKE 53281,2:REM" HINTERGRUND GRAU           <219>
921 :                                            <135>
922 REM   POKE792,193:REM" RESTORE UND           <137>
923 REM POKE788,52 :REM" STOP VERRIEGELN         <155>
924 :                                            <138>
925 DIM KW$(254):REM" KREUZWOERTER               <209>
926 DIM FW$(254):REM" FRAGEN DAZU                <022>
927 DIM FS$(255):REM" FRAGESTELLUNG              <137>
928 FG=1:OG=FG :REM" FELDGROESSE                 <202>
929 POKE 2,FG    :REM" UEBERGEBEN                <179>
930 :                                            <144>
931 POKE 785,0    :REM" USR-VEKTOR               <087>
932 POKE 786,192 :REM" SETZEN                    <210>
933 :                                            <147>
934 LS=0:REM" AUTOLADER AUS                      <087>
935 BS=1:REM" BLAUSPERRE EIN                     <227>
936 :                                            <150>
937 SYS 50016:REM" MATRIX LOESCHEN               <089>
938 :                                            <152>
940 REM SYS 51859 :REM" INIT. DRUCKTREIBER       <007>
942 :                                            <156>
944 REM" KONSTANTEN:                             <058>
945 :                                            <159>
946 LR=10   :REM" LINKER RAND DRUCKAUSG.         <178>
948 CZ=INT(130-LR*1.7) :REM" ZEILLAENGE         <091>
949 :                                            <163>
950 FOR N=1 TO 40    :REM" LEERSTRING FUER       <031>
952 :LL$=LL$+" " :REM" FORMATIERTE               <006>
954 NEXT             :REM" AUSGABE               <029>
955 :                                            <169>
956 S1=50176 :REM" ADRESSEN DER                  <224>
958 S2=50576 :REM" WORTFELD-                     <104>
960 S3=50976 :REM" SPEICHER                      <173>
962 S4=1186                                      <165>
963 :                                            <177>
964 DM=10000:REM" MAX. ANZ. LAEUFE               <003>
965 :                                            <179>
966 RETURN                                       <006>
967 :                                            <181>
969 :                                            <183>
970 REM" ROUTINEN LADEN                          <005>
971 REM" ******************************          <015>
972 :                                            <186>
975 FOR N=0 TO 3:S1=S1+PEEK(49152+N):NEXT       <237>
976 FOR N=0 TO 3:S2=S2+PEEK(51857+N):NEXT       <204>
977 :                                            <191>
978 IF S1<>483 THEN LOAD"SUCH.OBJ",8,1          <153>
979 REM IFS2<>494THENLOAD"DRUCK.OBJ",8,1        <045>
980 :                                            <194>
987 RETURN                                       <027>
997 :                                            <211>
998 :                                            <212>
```

Listing 1. »Raetsel« (Fortsetzung)

```
999 :                                            <213>
1000 REM" INPUTFORM ROUTINE                      <235>
1001 REM" ******************************         <172>
1002 :                                           <216>
1003 X$=CHR$(13):Y$=CHR$(20)                     <066>
1004 Z$=CHR$(34)                                 <200>
1005 :                                           <219>
1006 PRINT"{HOME,CYAN,RVSON}";T$;"?{RVOFF,
     2SPACE}";:POKE 198,0                        <129>
1007 :                                           <221>
1008 F$=""                                       <039>
1009 FOR N=1 TO L+1                              <027>
1010 :PRINT"{LEFT}";F$;"@";                      <141>
1011 :WAIT 198,1:GET F$:POKE 198,0               <016>
1012 :IF F$=X$THEN 1024                          <095>
1013 :IF F$=Y$AND N>1 THEN N=N-1:GOTO 1010       <135>
1014 :IF N>L THEN 1011                           <030>
1015 :IF M=0 THEN 1021                           <044>
1016 :IF F$=","OR F$=Z$THEN 1011                 <240>
1017 :IF F$>=" "AND F$<="9"THEN 1022             <052>
1018 :IF F$>="A"AND F$<="Z"THEN 1022             <034>
1021 :IF F$<"A"OR F$>"Z"THEN 1011                <193>
1022 NEXT                                        <014>
1023 :                                           <237>
1024 PRINT"{LEFT,SPACE,LEFT,HOME,RVSON}";T
     $;                                          <149>
1026 POKE 631,13:POKE 198,1:INPUT F$             <095>
1029 IF F$=X$THEN 1005                           <202>
1030 :                                           <246>
1097 RETURN                                      <139>
1098 :                                           <058>
1099 :                                           <059>
2000 REM" WORT POSITIONIEREN                     <096>
2001 REM" ******************************         <156>
2002 :                                           <200>
2004 PRINT"{HOME,DOWN,LIG.BLUE,RVSON,SPACE
     }(S) ENKRECHT{3SPACE}ODER";                 <229>
2005 PRINT"{3SPACE}(W) AAGERECHT{2SPACE}?
     "                                           <177>
2006 WAIT 203,63:GET F$                          <239>
2007 IF F$<>"S"AND F$<>"W"THEN 2006              <150>
2008 :                                           <206>
2009 PRINT"{HOME,DOWN,RVOFF}";LL$;               <062>
2010 PRINT"{HOME,DOWN,RVSON}BITTE POSITION
     ANFAHREN!"                                  <221>
2011 PRINT"{HOME,3DOWN,3RIGHT,RVOFF}";           <103>
2012 :                                           <210>
2013 FA=1186:P=FA:F=PEEK(FA)                     <198>
2014 X=0:Y=0:S=(F$="S"):POKE 140,ABS(S)          <158>
2015 MX=20+L*(S=0)+(S=-1)                        <146>
2016 MY=20+L*(S=-1)+(S=0)                        <034>
2017 :                                           <215>
2018 FOR N=0 TO 9999                             <237>
2019 POKE P,F:P=FA+X+Y*40                        <195>
2020 F=PEEK(P):POKE P,94:WAIT 198,1              <107>
2021 GET F$:IF F$=CHR$(13)THEN 2028              <211>
2022 IF F$="{DOWN}"THEN Y=Y-1*(Y<MY):NEXT        <174>
2023 IF F$="{RIGHT}"THEN X=X-1*(X<MX):NEXT       <143>
2024 IF F$="{UP}"THEN Y=Y+1*(Y>0):NEXT           <207>
2025 IF F$="{LEFT}"THEN X=X+1*(X>0):NEXT         <192>
2026 NEXT                                        <002>
2027 :                                           <225>
2028 POKE P,F                                    <100>
2029 PRINT"{HOME}";LL$;LL$;                      <211>
2030 :                                           <228>
2031 POKE 211,X :REM" CURSOR-POSITION            <218>
2032 POKE 214,Y :REM" SETZEN                     <028>
2033 :                                           <231>
2097 RETURN                                      <123>
2098 :                                           <042>
2099 :                                           <043>
3000 REM" DIRECTORY LISTEN                       <170>
3001 REM" ******************************         <140>
3003 :                                           <185>
3004 PRINT"{HOME}";LL$;"{HOME,CYAN}";            <140>
3005 PRINT"MIT{SPACE,RVSON}RETURN{RVOFF,SP
     ACE}AUSWAEHLEN "                            <027>
3006 PRINT"{UP}MIT{SPACE,RVSON}LEERTASTE{R
     VOFF,2SPACE}WEITER "                        <214>
3007 :                                           <189>
3008 OPEN 15,8,15,"IO":OPEN 2,8,2,"#"            <086>
3010 :                                           <192>
3012 T=18:S=1:F$=""                              <112>
3014 :                                           <196>
3016 FOR W=0 TO 99                               <205>
3018 :                                           <200>
3020 :PRINT#15,"B-R";2;0;T;S                     <161>
3022 :PRINT#15,"B-P";2;0                         <050>
3024 :                                           <206>
3026 :GET#2,X$:T=ASC(X$+CHR$(0))                 <180>
3028 :GET#2,X$:S=ASC(X$+CHR$(0))                 <178>
3030 :                                           <212>
3032 :FOR X=0 TO 7                               <082>
3034 : PRINT#15,"B-P";2;X*32+5:GET#2,F$          <216>
3038 : IF F$<>"±"THEN NEXT X:GOTO 3060           <243>
3042 : FOR Y=1 TO 15                             <016>
3044 :  GET#2,X$:F$=F$+X$                        <097>
3046 :  IF X$=CHR$(160)THEN Y=15                 <139>
3048 : NEXT Y                                    <143>
3050 : PRINT"{HOME}";TAB(24);"{RVSON}";          <068>
3052 : PRINT RIGHT$(F$,LEN(F$)-2);"{RVOFF}
     "                                           <198>
3053 : WAIT 203,64:WAIT 203,63:GET X$            <073>
3054 : IF X$=CHR$(13)THEN 3064                   <186>
3055 : PRINT"{HOME}";TAB(24);LEFT$(LL$,14)       <179>
3056 : F$=""                                     <081>
3057 :NEXT X                                     <136>
3058 :                                           <240>
3060 :IF T=0 THEN W=99                           <049>
3061 NEXT W                                      <207>
3062 :                                           <244>
3064 CLOSE 2:CLOSE 15                            <190>
3065 PRINT"{HOME}";LL$;LL$                       <025>
3066 :                                           <248>
3068 RETURN                                      <076>
3070 :                                           <252>
3071 :                                           <253>
4000 REM" SICHERHEITSABFRAGE                     <098>
4001 REM" ******************                     <046>
4002 :                                           <168>
4003 POKE 53280,2:POKE 53281,2                   <077>
4004 PRINT"{HOME}";LL$                           <054>
4006 PRINT"{HOME,CYAN,RVSON}";F$;": {RVOFF,
     SPACE}SICHER (J/N) ?"                       <212>
4008 WAIT 203,64:WAIT 203,63:GET F$              <227>
4010 IF F$<>"J"THEN PRINT"{HOME}";LL$            <114>
4012 :                                           <178>
4014 RETURN                                      <006>
4016 :                                           <182>
4018 :                                           <184>
4020 :                                           <186>
9000 REM"******************************          <135>
9002 REM"-{2SPACE}ENDE{3SPACE}RAETSELGENER
     ATOR{3SPACE}-                               <129>
9004 REM"******************************          <120>
© 64'er
```

Listing 1. Das Listing zum Kreuzworträtsel. Beachten Sie die Hinweise zum Eintippen auf Seite 16. Speichern Sie dieses Programm unter dem Namen »RAETSEL« auf Diskette. Vor dem Starten muß zuerst Listing 2 eingegeben und gespeichert werden.

```
10 PRINT"{CLR}";:T=TI+180                        <051>
11 PRINT"******************************"         <076>
12 PRINT"- LEXIKON-EDITOR{2SPACE}HH,30.5.8
   4 -                                           <156>
18 PRINT"******************************"         <064>
19 IF TI<T THEN 19                               <120>
20 :                                             <252>
22 :                                             <254>
24 REM" HAUPTPROGRAMM                            <072>
26 REM" *************                            <219>
28 :                                             <004>
30 GOSUB 900:REM" VORBEREITUNGEN                 <129>
32 :                                             <116>
34 GOSUB 600:REM" MENUE                          <167>
36 :                                             <012>
38 M=0                                           <243>
40 IF F$="{F1}"THEN     GOSUB 3006:GOTO 34       <028>
42 IF F$="{F3}"THEN     GOSUB 0300:GOTO 34       <054>
44 IF F$="{F5}"THEN CLR:GOSUB 0500:GOTO 34       <012>
46 IF F$="{F7}"THEN PRINT"{CLR}"                 <087>
47 :                                             <023>
48 PRINT " SICHER{11SPACE}(J/N) ?"               <162>
50 WAIT 203,64:POKE 198,0:WAIT 203,63            <110>
52 GET F$:IF F$<>"J"THEN 34                      <122>
53 :                                             <029>
54 PRINT CHR$(9);                                <106>
60 PRINT"{DOWN,SPACE}RAETSEL-PROGRAMM (J/N
   ) ?"                                          <177>
62 WAIT 203,64:POKE 198,0:WAIT 203,63            <122>
63 :                                             <039>
64 GET F$:IF F$="N"THEN 90                       <133>
```

Listing 2. »Editor«

```
66 IF F$<>"J"THEN 62                         <215>
68 PRINT"{CLR}LOAD"+CHR$(34);                <036>
70 PRINT"RAETSEL";CHR$(34);",8"              <198>
72 PRINT"{3DOWN}":PRINT"RUN"                 <116>
74 POKE 198,3:POKE 631,19                    <201>
76 POKE 632,13:POKE 633,13                   <156>
77 GOTO 97                                   <175>
88 :                                         <064>
90 POKE 792,71:REM" RESTORE UND              <205>
92 POKE 788,49:REM" STOP REPARIEREN          <168>
94 :                                         <070>
97 END                                       <099>
98 :                                         <074>
99 :                                         <075>
300 REM" DATEI AENDERN                       <153>
302 REM" *************                       <241>
304 :                                        <026>
306 GOSUB 700:REM" DATEI EINLESEN            <247>
307 :                                        <029>
308 EF=0:PRINT"{CLR}"                        <217>
309 PRINT" MIT{SPACE,RVSON}RETURN{RVOFF,SP
    ACE}AUSWAEHLEN"                          <127>
310 PRINT" MIT{SPACE,RVSON}LEERTASTE{RVOFF
    ,2SPACE}WEITER"                          <228>
311 PRINT" MIT{SPACE,RVSON}Q{RVOFF,7SPACE}
    ABBRECHEN"                               <220>
312 GOSUB 2000:REM" DATEI AUFLISTEN          <149>
313 :                                        <035>
314 PRINT"{DOWN,3SPACE,RVSON}FERTIG ?{RVOF
    F,SPACE}(J/N)"                           <166>
316 WAIT 203,64:POKE 198,0:WAIT 203,63       <122>
318 GET F$:IF F$="N"THEN 308                 <059>
320 IF F$<>"J"THEN 316                       <117>
322 :                                        <044>
323 IF EF=0 THEN 397                         <183>
324 GOSUB 800:REM" DATEI AUFZEICHNEN         <125>
395 :                                        <117>
397 RETURN                                   <201>
398 :                                        <120>
399 :                                        <121>
500 REM" DATEI EINGEBEN                      <110>
502 REM" **************                      <058>
503 :                                        <225>
504 PRINT"{CLR}";:GOSUB 900                  <025>
505 :                                        <227>
506 T$="{RVSON}WORTSCHATZ-NAME:"             <245>
507 L=14:M=1:F$="":GOSUB 1000                <146>
508 AF$="±"+CHR$(20)+F$                      <111>
510 :                                        <232>
519 PRINT                                    <113>
520 T$="{RVSON}FOLGE-WORTSCHATZ (Q=KEINEN)
    :"                                       <119>
521 L=14:M=1:F$="":GOSUB 1000                <162>
522 NF$="±"+CHR$(20)+F$                      <179>
524 :                                        <248>
525 PRINT"{CLR}":NR=0:EF=0                   <226>
527 :                                        <251>
528 REM" WOERTER EINGEBEN                    <097>
529 GOSUB 650                                <085>
530 :                                        <254>
532 IF F$="Q"THEN NR=NR-1:GOTO 539           <167>
534 IF NR<254 THEN NR=NR+1                   <215>
536 IF PEEK(210)=7 THEN PRINT"{CLR}"         <005>
537 GOTO 529                                 <054>
538 :                                        <006>
539 IF EF=0 THEN 597                         <147>
540 REM" DATEI AUF DISKETTE ABLEGEN          <077>
541 FG=NR:GOSUB 800                          <087>
592 :                                        <060>
597 RETURN                                   <147>
598 :                                        <066>
599 :                                        <067>
600 REM" MENUE                               <169>
602 REM" *****                               <033>
604 :                                        <072>
606 PRINT"{CLR}";:F$=""                      <204>
608 PRINT"{DOWN,SPACE,RVSON}F1{RVOFF,2SPAC
    E}DIRECTORY                              <197>
610 PRINT"{DOWN,SPACE,RVSON}F3{RVOFF,2SPAC
    E}DATEI AENDERN                          <012>
612 PRINT"{DOWN,SPACE,RVSON}F5{RVOFF,2SPAC
    E}DATEI ANLEGEN                          <052>
614 PRINT"{DOWN,SPACE,RVSON}F7{RVOFF,2SPAC
    E}ABBRECHEN                              <082>
616 :                                        <084>
618 WAIT 203,63:GET F$                       <121>
620 IF F$<CHR$(133)OR F$>CHR$(136)THEN 618   <142>
621 PRINT"{CLR}"                             <101>
622 :                                        <090>
647 RETURN                                   <197>
648 :                                        <116>
649 :                                        <117>
650 REM" KREUZWOERTER EINGEBEN               <204>
651 REM" *********************               <078>
652 :                                        <120>
653 N$=STR$(NR):N$=RIGHT$(N$,LEN(N$)-1)      <248>
654 T$="{RVSON}"+N$+". KREUZWORT{RVOFF,SPA
    CE}(Q=ENDE)"                             <062>
655 L=20:M=0:F$=KW$(NR):GOSUB 1000           <090>
656 IF F$="Q"THEN 697                        <065>
657 IF LEN(F$)<2 THEN PRINT"{3UP}";:GOTO 6
    55                                       <235>
658 KW$(NR)=F$:PRINT                         <030>
659 :                                        <127>
664 T$="{RVSON}FRAGESTELLUNG{RVOFF}"         <139>
666 L=35:M=1:F$=FW$(NR):GOSUB 1000           <214>
681 FW$(NR)=F$:PRINT                         <033>
682 :                                        <150>
683 EF=1                                     <071>
686 :                                        <154>
697 RETURN                                   <247>
698 :                                        <166>
699 :                                        <167>
700 REM" DATEI EINLESEN                      <204>
702 REM" **************                      <004>
703 :                                        <171>
705 PRINT"{CLR}"                             <185>
706 T$="{RVSON}WORTSCHATZ{RVOFF,SPACE}($=D
    IRECTORY)"                               <002>
708 L=14:M=1:F$="":GOSUB 1000                <093>
709 IF F$="$"THEN M=1:GOSUB 3000             <060>
710 IF F$=""THEN 705                         <217>
712 :                                        <180>
713 AF$="±"+CHR$(20)+F$                      <062>
714 OPEN 1,8,2,"0:"+AF$+",S,R"               <110>
716 :                                        <184>
718 INPUT#1,FG:REM" FELDGROESSE             <085>
719 :                                        <187>
720 REM" WOERTER/FRAGEN EINLESEN             <196>
721 FOR N=0 TO FG                            <017>
722 :INPUT#1,KW$(N),FW$(N)                   <198>
724 NEXT                                     <226>
726 :                                        <194>
728 INPUT#1,NF$:REM" FOLGE-WORTSCHATZ        <071>
730 :                                        <198>
732 CLOSE 1                                  <235>
796 :                                        <010>
797 RETURN                                   <093>
798 :                                        <012>
799 :                                        <013>
800 REM" DATEI AUFZEICHNEN                   <183>
802 REM" *****************                   <127>
803 :                                        <017>
805 PRINT"{CLR}"                             <031>
807 :                                        <021>
812 OPEN 15,8,15,"I":CLOSE 15                <226>
813 :                                        <027>
814 OPEN 1,8,2,"@0:"+AF$+",S,W"              <152>
816 :                                        <030>
818 PRINT#1,FG :REM" FELDGROESSE            <011>
819 :                                        <033>
820 FOR N=0 TO FG                            <118>
822 :PRINT#1,KW$(N):PRINT#1,FW$(N)           <217>
824 NEXT                                     <072>
826 :                                        <040>
828 PRINT#1,NF$:REM" NAECHSTER FILE          <066>
830 :                                        <044>
832 CLOSE 1                                  <081>
896 :                                        <110>
897 RETURN                                   <193>
898 :                                        <112>
899 :                                        <113>
900 REM" VORBEREITUNGEN                      <042>
902 REM" **************                      <206>
904 :                                        <118>
906 POKE 53280,6:REM" RAHMEN BLAU            <117>
908 POKE 53281,14:REM" GRUND HELLBLAU        <149>
909 PRINT CHR$(14);CHR$(8);CHR$(144)         <091>
910 :                                        <124>
912 POKE 792,193:REM" RESTORE UND            <038>
913 POKE 788,52:REM" STOP VERRIEGELN         <222>
914 :                                        <128>
915 DIM KW$(254):REM" KREUZWOERTER           <199>
916 DIM FW$(254):REM" FRAGEN DAZU            <012>
986 :                                        <200>
987 RETURN                                   <027>
```

Listing 2. »Editor« (Fortsetzung)

```
988 :                                                <202>
989 :                                                <203>
990 :                                                <204>
1000 REM" INPUTFORM ROUTINE                          <235>
1001 REM" *****************                          <070>
1002 :                                               <216>
1003 X$=CHR$(13):Y$=CHR$(20)                         <066>
1004 Z$=CHR$(34):POKE 198,0                          <250>
1005 :                                               <219>
1006 PRINT" ";T$:PRINT"{2DOWN,UP,SPACE}? "
     ;F$;" ";                                        <031>
1007 X=1:IF F$<>""THEN X=LEN(F$)+1                   <177>
1008 :                                               <222>
1009 F$="":FOR N=X TO L+1                            <003>
1010 :PRINT"{LEFT}";F$;"█";                          <141>
1011 :WAIT 198,1:GET F$:POKE 198,0                   <016>
1012 :IF F$=X$THEN 1025                              <127>
1013 :IF F$=Y$AND N>1 THEN N=N-1:GOTO 1010           <135>
1014 :IF N>L THEN 1011                               <030>
1015 :IF M=0 THEN 1021                               <044>
1016 :IF F$=","OR F$=Z$THEN 1011                     <240>
1017 :IF F$>=" "AND F$<="9"THEN 1022                 <052>
1018 :IF F$>="A"AND F$<="Z"THEN 1022                 <034>
1021 :IF F$<"A"OR F$>"Z"THEN 1011                    <193>
1022 NEXT                                            <014>
1023 :                                               <237>
1025 PRINT CHR$(20);:POKE 211,1                      <143>
1026 POKE 631,13:POKE 198,1:INPUT F$                 <095>
1030 IF F$=X$THEN PRINT"{UP,4RIGHT}";:GOTO
     1009                                            <242>
1036 :                                               <252>
1097 RETURN                                          <139>
1098 :                                               <058>
1099 :                                               <059>
2000 REM" DATEI AUFLISTEN                            <110>
2002 REM" ***************                            <037>
2004 :                                               <202>
2010 FOR NR=0 TO FG                                  <013>
2012 :PRINT"{DOWN,SPACE}->{RVSON}";KW$(NR)
     ;"{RVOFF}"                                      <032>
2014 :PRINT"{3SPACE}";FW$(NR)                        <242>
2016 :FOR W=0 TO 60:NEXT                             <157>
2017 :WAIT 198,1:GET F$:PRINT"{2UP,3SPACE,
     2DOWN}"                                         <200>
2018 :IF F$="Q"THEN 2029                             <052>
2019 :IF F$<>CHR$(13)THEN 2024                       <025>
2021 :PRINT"{CLR}":GOSUB 650                         <046>
2022 :PRINT"{CLR}":WAIT 203,64                       <197>
2024 :IF PEEK(210)=7 THEN PRINT"{CLR}"               <180>
2026 NEXT                                            <002>
2028 :                                               <226>
2029 T$="{RVSON}FOLGE-WORTSCHATZ"                    <071>
2030 PRINT"{3SPACE}";T$;":":PRINT" ->";NF$           <087>
2032 WAIT 203,64:POKE 198,0:WAIT 203,63              <058>
2034 :                                               <232>
2036 GET F$:IF F$<>CHR$(13)THEN 2097                 <230>
2037 F$=RIGHT$(NF$,LEN(NF$)-2)                       <173>
2038 T$=T$+" (Q=KEINEN):"                            <118>
2039 PRINT"{CLR}":GOSUB 1000:EF=1                    <133>
2040 NF$="±"+CHR$(20)+F$                             <171>
2096 :                                               <040>
2097 RETURN                                          <123>
2098 :                                               <042>
2099 :                                               <043>
3000 REM" DIRECTORY LISTEN                           <170>
3002 REM" ****************                           <027>
3003 :                                               <185>
3004 PRINT"{CLR}"                                    <198>
3005 PRINT" MIT{SPACE,RVSON}RETURN{RVOFF,S
     PACE}AUSWAEHLEN"                                <027>
3006 PRINT" MIT{SPACE,RVSON}LEERTASTE{RVOF
     F,2SPACE}WEITER"                                <128>
3007 PRINT                                           <059>
3008 :                                               <190>
3009 OPEN 15,8,15,"IO":OPEN 2,8,2,"#"                <087>
3010 :                                               <192>
3012 T=18:S=1:F$=""                                  <112>
3014 :                                               <196>
3016 FOR W=0 TO 99                                   <205>
3018 :                                               <200>
3020 :PRINT#15,"B-R";2;0;T;S                         <161>
3022 :PRINT#15,"B-P";2;0                             <050>
3024 :                                               <206>
3026 :GET#2,X$:T=ASC(X$+CHR$(0))                     <180>
3028 :GET#2,X$:S=ASC(X$+CHR$(0))                     <178>
3030 :                                               <212>
3032 :FOR X=0 TO 7                                   <082>
3034 : PRINT#15,"B-P";2;X*32+5                       <212>
3036 : GET#2,X$                                      <186>
3038 : IF X$<>"±"THEN NEXT X:GOTO 3060               <020>
3040 : GET#2,X$                                      <190>
3042 : FOR Y=1 TO 14                                 <014>
3044 :  GET#2,X$:F$=F$+X$                           <097>
3046 :  IF X$=CHR$(160)THEN Y=15                     <139>
3048 : NEXT Y                                        <143>
3050 : PRINT" ->{SPACE,RVSON}";F$;"{RVOFF}
     "                                               <145>
3051 : IF M=0 THEN 3056                             <102>
3053 : WAIT 203,64:WAIT 203,63:GET X$                <073>
3054 : IF X$=CHR$(13)THEN 3064                       <186>
3055 : PRINT"{UP,4SPACE}"                            <149>
3056 : F$=""                                         <081>
3057 :NEXT X                                         <136>
3058 :                                               <240>
3060 :IF T=0 THEN W=99                               <049>
3061 NEXT W                                          <207>
3062 :                                               <244>
3064 CLOSE 2:CLOSE 15                                <190>
3065 IF M=0 THEN WAIT 203,63                         <146>
3066 :                                               <248>
3068 RETURN                                          <076>
3070 :                                               <252>
3071 :                                               <253>
3072 :                                               <000>
9000 REM"*******************************             <135>
9002 REM" ENDE{5SPACE}LEXIKON-EDITOR{4SPA
     CE}                                             <102>
9004 REM"*******************************             <120>
```


Listing 2. »Editor« (Schluß).
Beachten Sie die Eingabehinweise auf Seite 16.

```
programm : such.obj        c000 c38a
-------------------------------------
c000 : 78 20 8b c0 f0 09 20 bc  d5     c1c0 : 20 f0 06 e6 1d a0 01 d0  33
c008 : c0 20 f0 c0 4c 15 c0 20  dd     c1c8 : d0 18 60 a5 50 a6 51 85  fc
c010 : 0f c1 20 64 c1 b0 13 a9  d6     c1d0 : 57 86 58 a4 1d f0 0e a9  fa
c018 : 00 85 8b 8d 00 dc ad 01  0f     c1d8 : 28 18 65 57 90 02 e6 58  b6
c020 : dc c9 ff f0 ea a0 01 4c  4f     c1e0 : 85 57 88 d0 f2 a9 14 a6  67
c028 : 87 c0 a5 8c f0 06 20 15  f4     c1e8 : 8c f0 06 38 e5 1d 4c f4  d7
c030 : c2 4c 37 c0 20 5f c2 b0  68     c1f0 : c1 38 e5 1c 85 62 38 60  d7
c038 : 2d e6 19 a5 02 c5 19 90  a7     c1f8 : a9 00 85 8d 85 8f 85 1b  d5
c040 : 11 a2 01 20 38 c1 20 47  87     c200 : a5 57 a6 58 85 59 86 5a  f7
c048 : c1 a5 62 c5 1a 90 ea 4c  98     c208 : a5 5b a6 5c 85 5d 86 5e  aa
c050 : 2a c0 a5 14 d0 0b a5 15  ec     c210 : a5 1c 85 1e 60 20 f8 c1  57
c058 : d0 05 a0 02 4c 87 c0 c6  a5     c218 : a4 1e b1 59 c9 a0 f0 08  d8
c060 : 15 c6 14 4c 17 c0 a5 8b  8c     c220 : a4 1b d1 65 d0 35 a4 1e  f8
c068 : d0 0c a5 8d f0 e4 a5 8e  43     c228 : b1 5d 29 0f c9 0e d0 04  0c
c070 : f0 04 a5 8f f0 dc ee f8  61     c230 : 85 8f f0 06 c9 07 d0 23  d8
c078 : 07 20 92 c2 a0 03 ad f8  57     c238 : 85 8d e6 1b a5 1b c5 1a  1f
c080 : 07 c9 ff d0 02 a0 04 58  6c     c240 : f0 1b a9 28 18 65 59 90  60
c088 : 4c a2 b3 20 f7 b7 a5 2f  48     c248 : 02 e6 5a 85 59 a9 28 18  b8
c090 : a6 30 85 4e 86 4f a0 00  df     c250 : 65 5d 90 02 e6 5e 85 5d  fa
c098 : 84 8b a9 07 18 65 4e 90  34     c258 : 18 90 bd 18 60 38 60 20  b4
c0a0 : 02 e6 4f 85 4e a9 a2 a2  9c     c260 : f8 c1 a4 1e b1 59 c9 a0  74
c0a8 : 04 85 50 86 51 a2 d8 85  ec     c268 : f0 08 a4 1b d1 65 d0 20  b5
c0b0 : 52 86 53 a5 1a c9 00 f0  a1     c270 : a4 1e b1 5d 29 0f c9 0e  8a
c0b8 : 02 85 8b 60 a5 d3 85 1c  b3     c278 : d0 04 85 8f f0 06 c9 01  06
c0c0 : a5 d6 85 1d 18 69 04 85  bd     c280 : d0 0e 85 8d e6 1e e6 1b  9b
c0c8 : d6 20 6c e5 a5 d1 85 57  34     c288 : a5 1b c5 1a d0 d4 38 60  c5
c0d0 : a5 d2 85 58 e6 57 e6 57  be     c290 : 18 60 20 f8 c1 a4 1b b1  10
c0d8 : 20 24 ea a5 f3 85 5b a5  9e     c298 : 65 a4 1e 91 59 a9 05 85  0b
c0e0 : f4 85 5c e6 5b e6 5b a5  30     c2a0 : 21 b1 5d 29 0f c9 0e d0  2f
c0e8 : 33 a6 34 85 65 86 66 60  11     c2a8 : 0c a9 07 85 21 a5 8c f0  4f
c0f0 : a4 1c b1 5b 29 0f c9 0e  c8     c2b0 : 04 a9 01 85 21 a5 21 91  61
c0f8 : f0 11 aa a5 8c d0 07 8a  51     c2b8 : 5d e6 1b a5 1b c5 1a f0  2e
c100 : c9 01 d0 09 f0 05 8a c9  94     c2c0 : 22 a5 8c f0 19 a9 28 18  a6
c108 : 07 d0 02 38 60 18 60 a9  9b     c2c8 : 65 59 90 02 e6 5a 85 59  48
c110 : 00 85 8c a5 02 20 58 c1  b1     c2d0 : a9 28 18 65 5d 90 02 e6  70
c118 : 85 19 a5 4e 85 63 a5 4f  06     c2d8 : 5e 85 5d 4c 95 c2 e6 1e  21
c120 : 85 64 a6 19 f0 03 20 38  bc     c2e0 : 4c 95 c2 a9 00 a2 c4 85  10
c128 : c1 20 47 c1 a9 ff 20 58  cf     c2e8 : 57 86 58 a4 1d f0 0e a9  12
c130 : c1 c9 80 90 02 85 8c 60  47     c2f0 : 14 18 65 57 90 02 e6 58  ba
c138 : a9 03 18 65 63 90 02 e6  a6     c2f8 : 85 57 88 d0 f2 85 59 a5  71
c140 : 64 85 63 ca d0 f2 60 a0  00     c300 : 58 85 5a a4 1c b1 59 a6  48
c148 : 00 b1 63 85 1a c8 b1 63  20     c308 : 8c f0 04 09 40 d0 02 09  d3
c150 : 85 65 c8 b1 63 85 66 60  ad     c310 : 04 91 59 e0 00 f0 0f 18  43
c158 : 85 61 ad 1b d4 c5 61 f0  3f     c318 : a5 59 69 90 85 59 a5 5a  45
c160 : 02 b0 f7 60 a2 13 86 1f  df     c320 : 69 01 85 5a d0 0d 18 a5  d8
c168 : 86 20 e8 8a 38 e5 1a a6  f2     c328 : 59 69 20 85 59 a5 5a 69  ee
c170 : 8c f0 06 85 20 a9 07 d0  b4     c330 : 03 85 5a ad f8 07 91 59  03
c178 : 04 85 1f a9 01 85 21 a5  48     c338 : e0 00 f0 17 a6 1a 18 ca  69
c180 : 1f 20 58 c1 85 1c a5 20  0e     c340 : f0 0d a9 14 65 57 90 03  fd
c188 : 20 58 c1 85 1d a5 52 a6  8b     c348 : e6 58 18 85 57 90 f0 a9  22
c190 : 53 85 5b 86 5c a4 1d f0  8f     c350 : 10 d0 08 98 18 65 1a a8  44
c198 : 0e a9 28 18 65 5b 90 02  ff     c358 : 88 a9 01 11 57 91 57 60  37
c1a0 : e6 5c 85 5b 88 d0 f2 a4  a5     c360 : a9 00 a2 c4 85 59 86 5a  3c
c1a8 : 1c b1 5b 29 0f c9 0e f0  f2     c368 : a0 00 8c f8 07 98 91 59  79
c1b0 : 1a c5 21 f0 16 c6 1c 10  3b     c370 : e6 59 d0 fa e6 5a a5 5a  23
c1b8 : ee a5 1f 85 1c a5 1d c5  e0     c378 : c9 c9 d0 f1 a9 ff 8d 0e  85
                                       c380 : d4 8d 0f d4 a9 81 8d 12  7a
                                       c388 : d4 60 ff 00 ff 00 ff 00  8c
```

Listing 3. »Such.obj.«. Geben Sie das Maschinenprogramm mit dem MSE ein (Seite 18).

»Happysynth«, der Traum eines jeden Musikers

Daß der Commodore 64 hervorragende Musikqualitäten besitzt, ist längst kein Geheimnis mehr. Schließen Sie ihn an Ihre Stereoanlage an, und tippen Sie unseren Supersynthesizer ein. Sie werden Ihren Ohren nicht trauen.

Zu Anfang wollen wir einige Tips zur Hardware geben. Um den C 64 professionell in einer Band einzusetzen, wird die schlechte Qualität des Fernsehlautsprechers kaum ausreichen. Vorteilhafter ist es, den C 64 mit einem Verstärker zu betreiben. Das Tonsignal kann an der Audio/Video-Buchse des Commodore abgegriffen werden. Um nun den Computer an einen Verstärker anzuschließen, verbindet man einfach Pin 3 der Audio/Video-Buchse über ein abgeschirmtes Kabel mit dem Eingang des Verstärkers. Weniger bastelgeübten Lesern ist ein handelsübliches, fünfpoliges Stereo-Überspielkabel zu empfehlen. Wer seinen C 64 mit einem Monitor betreibt, muß sich einen Zwischenstecker für die Audio/Video-Buchse basteln. Sie werden erstaunt sein, wie gut sich Ihr Commodore an einer guten Verstärkeranlage anhört.

Ebenso läßt sich an der Audio/Video-Buchse ein Tonsignal einspeisen (Pin 5). Dieses wird durch Setzen von Bit 3 im Register 23 über den Filter geleitet. Dadurch kann der SID-Chip auch als Effektgerät eingesetzt werden. So lassen sich Phasing-ähnliche Effekte erzielen, indem im SID ein Notchfilter (Hochpaß und Tiefpaß) realisiert wird, dessen Grenzfrequenz moduliert wird. Eine Modulation läßt sich mit Stimme 3 sehr einfach erreichen. Man bedient sich des Leseregisters, welches nur für diesen Oszillator existiert.

Als maximale Spannung kann der C 64 volle 3 V_{ss} verarbeiten. Wer also ein Mikrofon am C 64 betreiben möchte, muß das Signal mittels eines Vorverstärkers dem Computer anpassen. Auch hier ist unbedingt ein abgeschirmtes Kabel zu verwenden. Die Anschlußbelegung der Audio/Video-Buchse können Sie Ihrem Handbuch (Seite 142) entnehmen.

»Happysynth« (Listing 1) lädt nach dem Start erst zwei Maschinenprogramme, sowie die Daten für Sprites nach (Listing 2, 3 und 4). Die Maschinenprogramme beinhalten die Joystickabfrage, den Modulationsteil und die als Interrupt eingebundene Tastaturabfrage. Das heißt, es kann jederzeit, auch während der Parameteränderung, auf der Tastatur gespielt werden, da die Soundeinstellung ausschließlich mit dem Joystick gemacht wird. Drückt man nun den Feuerknopf, kann es losgehen.

Man befindet sich dann auf der ersten von drei Bildschirmseiten. Die am Bildschirm sichtbare Hand läßt sich mit Hilfe des Joysticks in jede Richtung bewegen. Durch erneutes Drücken des Feuerknopfes kann der jeweilige Parameter eingestellt, beziehungsweise verändert werden. Soll beispielsweise bei Oszillator 1 eine Sägezahnwelle eingeschaltet werden, so muß man lediglich die Hand auf das Sägezahn-Symbol steuern und den Knopf am Joystick drücken. Sogleich wird das Symbol grün, weil damit der Oszillator auf diese Wellenform umgestellt wurde. Drückt man an gleicher Stelle den Knopf ein weiteres Mal, leuchtet das Symbol wieder rot als Zeichen dafür, daß der Oszillator keinen Sägezahn-Ton mehr erzeugt. Diese Farben wurden im gesamten »Happysynth« beibehalten: Grün bedeutet immer, daß irgendeine Funktion oder ein Parameter eingeschaltet ist, rot signalisiert, daß die Funktion, beziehungsweise der Parameter ausgeschaltet ist. Bei Parametern, die sich nicht nur ein- und ausschalten lassen, sondern stufenlos reguliert werden, bedient man sich ebenfalls des Joysticks. Will man zum Beispiel die Pulsbreite einer Stimme verändern, so bewegt man die Hand zu der entsprechenden Stelle am Bildschirm und drückt den Feuerknopf.

Nun steht im Feld »Joystick-Informationen« in welcher Abstufung sich die angewählten Parameter verändern lassen. Im Falle der Pulsbreite sind es ±4 Prozent in Y-Richtung, was einer Grobeinstellung entspricht, sowie ±1 Prozent in X-Richtung, um die Pulsweite fein einzustellen. Bei der Einstellung der Filtergrenzfrequenz wird nur zwischen Grob- (Symbol auf dem Bildschirm: +++) und Feineinstellung (Symbol: +) unterschieden, da hier Zahlen wenig sinnvoll sind. Hat man in der ersten Bildschirmseite alle gewünschten Funktionen eingestellt, kann man auf der zweiten Bildschirmseite mit der Einstellung des »Synthesizers« fortfahren. Die zweite Seite wird über das Feld »Hüllkurve-Modulation« angewählt. Also: Die Hand auf das Feld steuern und Knopf drücken. Nun befinden Sie sich auf der zweiten Seite. Hier läßt sich die Hüllkurve für jede Stimme getrennt eingeben. Da bei Attack, Decay und Release die Abstufung logarithmisch in Millisekunden erfolgt, erscheint auch bei diesen Parametern im Joystick-Feld keine Zahlenangabe.

ADSR und Modulationen

Neben den ADSRs werden auf dieser Seite die Modulationen eingestellt. Jeder der drei Oszillatoren läßt sich in der Frequenz und in der Pulsweite modulieren. Hierfür stehen ein eigener, langsam schwingender Oszillator (LFO = Low Frequency Oscillator) oder die dritte Stimme zur Verfügung. Die Wellenformen des LFOs werden wieder mit der auf dem Bildschirm dargestellten Hand umgeschaltet. Die Intensität der Modulation ist für jeden Oszillator getrennt einstellbar. Wird anstatt des LFO Stimme 3 für Modulationszwecke herangezogen, so wird mit der aktuellen Wellenform von DCO 3 moduliert. Wurde beispielsweise auf Seite 1 für DCO 3 Rauschen eingeschaltet, so werden alle angewählten Parameter durch Zufallswerte beeinflußt. Einen weiteren interessanten Effekt bietet die Beeinflussung des Filters mit Hilfe des ADSR (»Envelope Follow«). Wird bei dieser Funktion die Intensität mit Hilfe des Steuerknüppels größer Null gesetzt, so folgt die Grenzfrequenz des Filters dem Hüllkurvenverlauf von ADSR 3. Je größer der bei »Envelope Follow« eingegebene Wert ist, um so stärker folgen die Werte des Filters dem Hüllkurvenverlauf. Um im Modulationsteil gute Ergebnisse zu erzielen, benötigt man etwas Übung und Fingerspitzengefühl. Hat man einen Sound gemixt, so will man ihn natürlich auch speichern, um ihn zu einem späteren Zeitpunkt wieder verwenden zu können. Die Routinen dazu werden auf Seite 3 aufgerufen. Sie wird über das Feld »Tastatur-Disk-Menü« angesteuert. Auf dieser Seite sind alle Aufrufe für die wichtigen Diskettenfunktionen untergebracht. Sie werden wie üblich ausgewählt.

Sie bewegen die Hand in diesem Fall also auf das Feld »Sound speichern«. Sogleich verschwindet die Tastatur im unteren Bildschirmbereich, um für die Eingabe des Soundnamens Platz zu machen. Hier wird zum ersten Mal die Tastatur nicht zum Spielen, sondern zur Eingabe des Soundnamens verwendet. Die Länge des Namens ist auf zwölf Zeichen beschränkt — längere Namen werden nicht angenommen. Durch

Drücken der »INST-DEL«-Taste wird der Name gelöscht um ihn neu einzugeben. Nach Drücken der »RETURN«-Taste werden die Klangparameter abgespeichert. Es können auf einer Diskette maximal Daten für 33 Sounds gespeichert werden — was darüber hinausgeht wird beim Laden einfach ignoriert! Listing 5 enthält sechs Demo-Sounds. Nach dem Starten mit »RUN« werden die Sounds in einer vom Hauptprogramm lesbaren Form auf Diskette gespeichert. Die Leserservice-Diskette enthält neben diesen noch acht weitere Sounds zu Versuchszwecken. Wollen Sie einen Ihrer Klänge wieder hören, so steuern Sie die Hand auf das Feld »Sound laden« und drücken den Feuerknopf. Nun wird eine Liste aller auf der Diskette vorhandenen Dateien angezeigt. Mit Hilfe des Joysticks wählen Sie den gewünschten Klang aus und drücken wieder den Feuerknopf — schon werden die Parameter geladen! Falls Sie die Funktion »Sound laden« ein weiteres Mal anwählen, wird die Soundliste sofort angezeigt, vorausgesetzt Sie haben die Diskette nicht gewechselt.

(Bernhard Carli/Christian Spitzner/tr)

```
100 GOSUB 7080:REM TITELBILD                    <105>
110 GOSUB 9030:REM MPG LADEN                    <144>
120 POKE 56,152:CLR                             <101>
130 GOSUB 9130:REM INITIALISIEREN               <073>
140 GOSUB 7430:REM START                        <140>
150 :                                           <126>
160 POKE V+21,0:POKE V+1,70                     <092>
170 ON PA GOTO 190,300,410                      <209>
180 PA=NR:GOTO 160                              <232>
190 GOSUB 8020:REM SEITE 1                      <244>
200 POKE V+16,6:POKE V+2,36:POKE V+4,36:PO
    KE V+3,109:POKE V+5,151                     <086>
210 POKE V,60:POKE V+21,7:POKE V+40,2:POKE
    V+41,2                                      <004>
220 SYS 49152                                   <022>
230 NR=PEEK(40959)                              <041>
240 ON NR GOTO 180,180,180,220,220,220,101
    0,1020,1040,1090,1100,1120,1170,1180        <080>
250 ON NR -14 GOTO 1200,220,220,1030,1260,
    1110,1330,1190,1400,220,220                 <131>
260 ON NR-25 GOTO 1050,1130,1210,1060,1140
    ,1220,220,220,220,1250,1320,1390            <013>
270 ON NR-37 GOTO 1250,1320,1390,1310,1380
    ,1450,2480,2950,2950,220,220,220            <235>
280 ON NR-49 GOTO 2510,2510,2510,2570,2570
    ,220,220,220,2520,2710,2710                 <115>
290 GOTO 220                                    <004>
300 GOSUB 8400:REM SEITE 2                      <164>
310 POKE V+21,7:POKE V+40,2:POKE V+41,2         <033>
320 SYS 49152                                   <124>
330 NR=PEEK(40959)                              <143>
340 ON NR GOTO 180,180,180,320,320,320,127
    0,1270,1270,1340,1340,1340                  <212>
350 ON NR-12 GOTO 1410,1410,1410,320,320,1
    280,1280,1350,1350,1420,1420                <069>
360 ON NR-23 GOTO 320,320,1290,1360,1430,3
    20,320,320,1300,1370,1440                   <072>
370 ON NR-34 GOTO 320,320,320,1070,1150,12
    30,1080,1160,1240,320,3170,3170             <123>
380 ON NR-46 GOTO 320,320,320,1460,1470,14
    80,2840,2840,320,320,320                    <004>
390 ON NR-57 GOTO 1490,3060,3060                <102>
400 GOTO 320                                    <122>
410 GOSUB 8740:REM SEITE 3                      <002>
420 POKE V+21,7:POKE V+40,2:POKE V+41,2         <143>
430 SYS 49152                                   <234>
440 NR=PEEK(40959)                              <253>
450 ON NR GOTO 180,180,180,4880,3700,4200       <226>
460 IF NR=>7 AND NR <=28 THEN 3570              <109>
470 GOTO 430                                    <216>
1000 REM +++ SPRUNGTABELLE +++                  <015>
1010 S=0:P=241:GOTO 1510                        <147>
1020 S=0:P=244:GOTO 1540                        <043>
1030 S=0:P=321:GOTO 1550                        <168>
1040 S=0:P=247:GOTO 1560                        <078>
1050 S=0:P=401:GOTO 1570                        <186>
1060 S=0:P=441:GOTO 1580                        <077>
1070 S=0:TA=9:GOTO 3470                         <013>
1080 S=0:TA=9:GOTO 3280                         <024>
1090 S=1:P=252:GOTO 1510                        <251>
1100 S=1:P=255:GOTO 1540                        <147>
1110 S=1:P=332:GOTO 1550                        <014>
1120 S=1:P=258:GOTO 1560                        <180>
1130 S=1:P=412:GOTO 1570                        <032>
1140 S=1:P=452:GOTO 1580                        <179>
1150 S=1:TA=20:GOTO 3470                        <244>
1160 S=1:TA=20:GOTO 3280                        <254>
1170 S=2:P=263:GOTO 1510                        <097>
1180 S=2:P=266:GOTO 1540                        <249>
1190 S=2:P=343:GOTO 1550                        <116>
1200 S=2:P=269:GOTO 1560                        <026>
1210 S=2:P=423:GOTO 1570                        <134>
1220 S=2:P=463:GOTO 1580                        <025>
1230 S=2:TA=31:GOTO 3470                        <090>
1240 S=2:TA=31:GOTO 3280                        <100>
1250 S=0:TA=9:GOTO 1640                         <033>
1260 S=0:TA=8:GOTO 1820                         <233>
1270 S=0:TA=3:GOTO 1960                         <010>
1280 S=0:TA=3:GOTO 2070                         <069>
1290 S=0:TA=9:GOTO 2180                         <188>
1300 S=0:TA=3:GOTO 2300                         <248>
1310 S=0:TA=8:GOTO 2410                         <108>
1320 S=1:TA=20:GOTO 1640                        <095>
1330 S=1:TA=19:GOTO 1820                        <011>
1340 S=1:TA=14:GOTO 1960                        <067>
1350 S=1:TA=14:GOTO 2070                        <171>
1360 S=1:TA=20:GOTO 2180                        <104>
1370 S=1:TA=14:GOTO 2300                        <252>
1380 S=1:TA=19:GOTO 2410                        <219>
1390 S=2:TA=31:GOTO 1640                        <187>
1400 S=2:TA=30:GOTO 1820                        <065>
1410 S=2:TA=25:GOTO 1960                        <159>
1420 S=2:TA=25:GOTO 2070                        <007>
1430 S=2:TA=31:GOTO 2180                        <196>
1440 S=2:TA=25:GOTO 2300                        <088>
1450 S=2:TA=30:GOTO 2410                        <017>
1460 P=842:L=1:MW=1:GOTO 3400                   <081>
1470 P=845:L=1:MW=0:GOTO 3400                   <156>
1480 P=848:L=1:MW=2:GOTO 3400                   <104>
1490 P=922:L=7:MW=3:GOTO 3400                   <088>
1500 :                                          <206>
1510 REM +++ WELLENFORMEN/SYNC/RING +++         <059>
1520 :                                          <226>
1530 DE(S)=1-DE(S):L=1:GOTO 1590                <001>
1540 SZ(S)=1-SZ(S):L=1:GOTO 1590                <213>
1550 RE(S)=1-RE(S):L=1:GOTO 1590                <135>
1560 RA(S)=1-RA(S):L=2:GOTO 1590                <085>
1570 RI(S)=1-RI(S):L=9:GOTO 1590                <251>
1580 SY(S)=1-SY(S):L=9                          <101>
1590 GOSUB 9820                                 <022>
1600 WF(S)=SY(S)*2+RI(S)*4+DE(S)*16+SZ(S)*
     32+RE(S)*64+RA(S)*128                      <126>
1610 POKE WF+S,WF(S)+8:POKE WF+S,WF(S)          <023>
1620 GOTO 220                                   <064>
1630 :                                          <082>
1640 REM +++ OKTAVE +++                         <212>
1650 :                                          <102>
1660 GOSUB 9430                                 <060>
1670 PRINT"{HOME,8DOWN,LIG.GREEN}"TAB(34)"
     +OKT.{3DOWN,5LEFT}+#{2DOWN,SPACE}-#{3
     DOWN,5LEFT}-OKT."                          <124>
1680 WERT=OK(S):JO=10:GOSUB 9530                <112>
1690 IF JO=0 THEN OK(S)=OK(S)-12                <246>
1700 IF JO=20 THEN OK(S)=OK(S)+12               <011>
1710 IF JO=9 THEN OK(S)=OK(S)-1                 <006>
1720 IF JO=11 THEN OK(S)=OK(S)+1                <211>
1730 IF OK(S)<0 OR OK(S)>=48 THEN OK(S)=WE
     RT                                         <211>
1740 POKE WF+3+S,OKT(S)                         <190>
1750 PRINT"{HOME,13DOWN,WHITE}"TAB(TA)INT(
     OKT(S)/12)                                 <228>
1760 PRINT TAB(TA)N$(INT(OK(S)-(INT(OKT(S)
     /12))*12))                                 <227>
1770 IF BU THEN 1680                            <126>
1780 PRINT"{2UP,PURPLE}"TAB(TA)INT(OKT(S)/
     12)                                        <197>
1790 PRINT TAB(TA)N$(INT(OK(S)-(INT(OKT(S)
     /12))*12))                                 <001>
1800 GOSUB 9440:GOSUB 9460:GOTO 220             <222>
1810 :                                          <008>
1820 REM +++ PULSE +++                          <013>
1830 :                                          <028>
1840 GOSUB 9430                                 <242>
1850 PRINT"{HOME,8DOWN,LIG.GREEN}"TAB(35)"
     +4{3DOWN,3LEFT}+1{2DOWN,SPACE}-1{3DOW
     N,3LEFT}-4"                                <147>
```

Listing 1. Das Hauptprogramm »Happysynth«. Bitte mit dem Checksummer V3 auf Seite 16 eingeben.

```
1860 WERT=PU(S)/16:JO=WERT                       <096>
1870 GOSUB 9530:PU(S)=JO*16                      <120>
1880 P1=INT(PU(S)/256)                           <047>
1890 POKE WF+49+S,PU(S)-P1*256:POKE WF+52+
     S,P1                                        <068>
1900 POKE SI+3+S*7,P1:POKE SI+2+S*7,PU(S)-
     P1*256                                      <015>
1910 PRINT"{HOME,8DOWN,WHITE}"TAB(TA)RIGHT
     $(STR$(INT(PU(S)/40.95)),2);"%"             <140>
1920 IF BU THEN 1870                             <022>
1930 PRINT"{UP,PURPLE}"TAB(TA)RIGHT$(STR$(
     INT(PU(S)/40.95)),2);"%"                    <125>
1940 GOSUB 9440:GOSUB 9460:GOTO 220              <106>
1950 :                                           <148>
1960 REM +++ ATTACK +++                          <252>
1970 :                                           <168>
1980 GOSUB 9430:GOSUB 9470                       <038>
1990 WERT=A(S):JO=WERT*16                        <245>
2000 GOSUB 9530:A(S)=INT(JO/16)                  <167>
2010 POKE SI+5+S*7,A(S)*16+D(S)                  <193>
2020 PRINT"{HOME,6DOWN,RVSON,GREEN}"TAB(TA
     )AD$(A(S))                                  <116>
2030 IF BU THEN 2000                             <185>
2040 PRINT"{UP,RVOFF,LIG.BLUE}"TAB(TA)AD$(
     A(S))                                       <170>
2050 GOSUB 9440:GOSUB 9460:GOTO 320              <090>
2060 :                                           <004>
2070 REM +++ DECAY +++                           <033>
2080 :                                           <124>
2090 GOSUB 9430:GOSUB 9470                       <150>
2100 WERT=D(S):JO=WERT*16                        <231>
2110 GOSUB 9530:D(S)=INT(JO/16)                  <026>
2120 POKE SI+5+S*7,A(S)*16+D(S)                  <049>
2130 PRINT"{HOME,8DOWN,RVSON,GREEN}"TAB(TA
     )R$(D(S))                                   <011>
2140 IF BU THEN 2110                             <170>
2150 PRINT"{UP,RVOFF,LIG.BLUE}"TAB(TA)R$(D
     (S))                                        <056>
2160 GOSUB 9440:GOSUB 9460:GOTO 320              <200>
2170 :                                           <114>
2180 REM +++ SUSTAIN +++                         <178>
2190 :                                           <134>
2200 GOSUB 9430                                  <092>
2210 PRINT"{HOME,8DOWN,LIG.GREEN}"TAB(36)"
     +1{2LEFT,8DOWN}-1"                          <090>
2220 WERT=S(S):JO=WERT*16                        <230>
2230 GOSUB 9530:S(S)=INT(JO/16)                  <161>
2240 POKE SI+6+S*7,S(S)*16+R(S)                  <235>
2250 PRINT"{HOME,10DOWN,RVSON,GREEN}"TAB(T
     A-6);"VOL.: ";RIGHT$(STR$(S(S)),2)          <038>
2260 IF BU THEN 2230                             <165>
2270 PRINT"{UP,RVOFF,LIG.BLUE}"TAB(TA-6);"
     VOL.: ";RIGHT$(STR$(S(S)),2)                <179>
2280 GOSUB 9440:GOSUB 9460:GOTO 320              <064>
2290 :                                           <234>
2300 REM +++ RELEASE +++                         <125>
2310 :                                           <000>
2320 GOSUB 9430:GOSUB 9470                       <126>
2330 WERT=R(S):JO=WERT*16                        <214>
2340 GOSUB 9530:R(S)=INT(JO/16)                  <016>
2350 POKE SI+6+S*7,S(S)*16+R(S)                  <091>
2360 PRINT"{HOME,12DOWN,RVSON,GREEN}"TAB(T
     A)R$(R(S))                                  <218>
2370 IF BU THEN 2340                             <150>
2380 PRINT"{UP,RVOFF,LIG.BLUE}"TAB(TA)R$(R
     (S))                                        <039>
2390 GOSUB 9440:GOSUB 9460:GOTO 320              <176>
2400 :                                           <090>
2410 REM +++ FILTER +++                          <219>
2420 :                                           <110>
2430 FE(S)=1-FE(S):L=2                           <249>
2440 POKE SI+23,RS*16+FE(0)+FE(1)*2+FE(2)*
     4                                           <177>
2450 PRINT"{HOME,16DOWN}"TAB(TA);FI$(FE(S)
     )                                           <081>
2460 GOTO 220                                    <142>
2470 :                                           <160>
2480 REM +++ HOCH- BAND- TIEFPASS +++            <074>
2490 :                                           <180>
2500 HP=1-HP:P=763:GOTO 2530                     <241>
2510 BP=1-BP:P=843:GOTO 2530                     <176>
2520 TP=1-TP:P=923                               <097>
2530 L=7:GOSUB 9820                              <207>
2540 POKE SI+24,15+HP*64+BP*32+TP*16             <199>
2550 GOTO 220                                    <232>
2560 :                                           <252>
2570 REM +++ FILTERFREQUENZ +++                  <014>
2580 :                                           <016>
2590 GOSUB 9430                                  <230>
2600 PRINT"{HOME,8DOWN,LIG.GREEN}"TAB(35)"
     +++{3DOWN,4LEFT}+{2DOWN,3SPACE}-{3DOW
     N,4LEFT}---"                                <002>
2610 WERT=30+CO*5.8:JO=WERT/46.4                 <000>
2620 GOSUB 9530:IF JO<.65 THEN JO=.65            <030>
2630 IF JO<>WERT/46.4 THEN CO=INT((JO*46.4
     -30)/5.8)                                   <102>
2640 C1=INT(CO/8)                                <089>
2650 POKE SI+22,C1:POKE SI+21,CO-C1*8:POKE
      WF+16,C1                                   <087>
2660 PRINT"{HOME,21DOWN,WHITE}"TAB(22);RIG
     HT$("{3SPACE}"+STR$(INT(CO*5.8+30)),5
     )                                           <241>
2670 IF BU THEN 2620                             <068>
2680 PRINT"{UP,PURPLE}"TAB(22);RIGHT$("{3S
     PACE}"+STR$(INT(CO*5.8+30)),5)              <255>
2690 GOSUB 9440:GOSUB 9460:GOTO 220              <094>
2700 :                                           <136>
2710 REM +++ FILTERRESONANZ +++                  <044>
2720 :                                           <156>
2730 GOSUB 9430                                  <114>
2740 PRINT"{HOME,8DOWN,LIG.GREEN}"TAB(36)"
     +1"                                         <174>
2750 PRINT"{7DOWN}"TAB(36)"-1"                   <226>
2760 WERT=RS:JO=WERT*16                          <002>
2770 GOSUB 9530:RS=INT(JO/16)                    <235>
2780 PRINT"{HOME,23DOWN,WHITE}"TAB(22);RIG
     HT$(STR$(RS),2)                             <153>
2790 POKE SI+23,RS*16+FE(0)+FE(1)*2+FE(2)*
     4                                           <017>
2800 IF BU THEN 2770                             <075>
2810 PRINT"{UP,PURPLE}"TAB(22);RIGHT$(STR$
     (RS),2)                                     <200>
2820 GOSUB 9440:GOSUB 9460:GOTO 220              <226>
2830 :                                           <012>
2840 REM +++ ENVELOPE-FOLLOW +++                 <197>
2850 :                                           <032>
2860 GOSUB 9430:GOSUB 9450                       <142>
2870 WERT=PEEK(WF+17):JO=WERT                    <082>
2880 GOSUB 9530:EF=JO                            <024>
2890 POKE WF+17,EF                               <231>
2900 PRINT"{HOME,21DOWN,WHITE}"TAB(30);RIG
     HT$(" "+STR$(EF),3)                         <253>
2910 IF BU THEN 2880                             <061>
2920 PRINT"{UP,PURPLE}"TAB(30);RIGHT$(" "+
     STR$(EF),3)                                 <151>
2930 GOSUB 9440:GOSUB 9460:GOTO 320              <208>
2940 :                                           <122>
2950 REM +++ VERSTIMMUNG +++                     <175>
2960 :                                           <142>
2970 GOSUB 9430:GOSUB 9450                       <252>
2980 WERT=VS:JO=WERT                             <020>
2990 GOSUB 9530:VS=JO:IF VS>99 THEN VS=99:
     JO=99                                       <242>
3000 POKE WF+6,VS                                <062>
3010 PRINT"{HOME,18DOWN,WHITE}"TAB(30)RIGH
     T$(STR$(VS),2)                              <013>
3020 IF BU THEN 2990                             <044>
3030 PRINT"{UP,PURPLE}"TAB(30)RIGHT$(STR$(
     VS),2)                                      <180>
3040 GOSUB 9440:GOSUB 9460:GOTO 1940             <012>
3050 :                                           <232>
3060 REM +++ FILTER-MODULATION +++               <178>
3070 :                                           <252>
3080 GOSUB 9430:GOSUB 9450                       <108>
3090 WERT=PEEK(WF+19):JO=WERT                    <113>
3100 GOSUB 9530:MF=JO                            <254>
3110 POKE WF+19,MF                               <214>
3120 PRINT"{HOME,23DOWN,WHITE}"TAB(30);RIG
     HT$(" "+STR$(MF),3)                         <063>
3130 IF BU THEN 3100                             <207>
3140 PRINT"{UP,PURPLE}"TAB(30);RIGHT$(" "+
     STR$(MF),3)                                 <181>
3150 GOSUB 9440:GOSUB 9460:GOTO 320              <174>
3160 :                                           <088>
3170 REM +++ MODULATION: SPEED +++               <100>
3180 :                                           <108>
3190 GOSUB 9430:GOSUB 9450                       <218>
3200 WERT=PEEK(WF+34):JO=WERT                    <094>
3210 GOSUB 9530:SP=JO                            <134>
3220 POKE WF+34,SP                               <118>
3230 PRINT"{HOME,18DOWN,WHITE}"TAB(30);RIG
     HT$(" "+STR$(SP),3)                         <119>
```

```
3240 IF BU THEN 3210                          <191>
3250 PRINT"{UP,PURPLE}"TAB(30);RIGHT$(" "+
     STR$(SP),3)                              <244>
3260 GOSUB 9440:GOSUB 9460:GOTO 320           <028>
3270 :                                        <198>
3280 REM +++ PWM-MODULATION +++               <194>
3290 :                                        <218>
3300 GOSUB 9430:GOSUB 9450                    <072>
3310 WERT=PEEK(WF+40+S):JO=INT(WERT/2.5757
     )                                        <189>
3320 GOSUB 9530:IF JO>99 THEN JO=99           <003>
3330 PW(S)=INT(JO*2.5757)+1                   <025>
3340 POKE WF+40+S,PW(S)                       <128>
3350 PRINT"{HOME,16DOWN,WHITE}"TAB(TA);RIG
     HT$(STR$(INT(PW(S)/2.5757)),2)           <120>
3360 IF BU THEN 3320                          <204>
3370 PRINT"{UP,PURPLE}"TAB(TA);RIGHT$(STR$
     (INT(PW(S)/2.5757)),2)                   <107>
3380 GOSUB 9440:GOSUB 9460:GOTO 320           <150>
3390 :                                        <064>
3400 REM +++ MODULATION-WELLENFORM +++        <167>
3410 :                                        <084>
3420 FOR I=842 TO 849:POKE FR+I,2:POKE FR+
     I+80,2:NEXT                              <064>
3430 FOR I=0 TO L:POKE FR+P+I,5:NEXT          <248>
3440 POKE WF+33,MW                            <227>
3450 GOTO 320                                 <124>
3460 :                                        <134>
3470 REM +++ VIBRATO +++                      <212>
3480 :                                        <154>
3490 GOSUB 9430:GOSUB 9450                    <008>
3500 WERT=PEEK(WF+64+S):JO=WERT               <172>
3510 GOSUB 9530:VI(S)=JO:IF VI(S)>99 THEN
     VI(S)=99:JO=99                           <217>
3520 POKE WF+64+S,VI(S)                       <005>
3530 PRINT"{HOME,14DOWN,WHITE}"TAB(TA);RIG
     HT$(STR$(VI(S)),2)                       <000>
3540 IF BU THEN 3510                          <110>
3550 PRINT"{UP,PURPLE}"TAB(TA);RIGHT$(STR$
     (VI(S)),2)                               <211>
3560 GOSUB 9440:GOSUB 9460:GOTO 320           <074>
3570 :                                        <244>
3580 REM +++ ENDE +++                         <193>
3590 :                                        <010>
3600 POKE V+21,0:POKE 198,0                   <214>
3610 PRINT"{HOME,7DOWN,RIGHT,25SPACE}"        <224>
3620 PRINT"{RIGHT,LIG.GREEN,7SPACE}SIND SI
     E SICHER ???"                            <199>
3630 PRINT"{RIGHT,25SPACE}"                   <214>
3640 GET A$:IF A$<>"J"AND A$<>"N"THEN 3640 <046>
3650 IF A$="N"THEN 410                        <170>
3660 POKE 53280,14:POKE 53281,6:POKE 198,0
     :POKE 808,237:POKE 2,0                   <254>
3670 PRINT"{CLR,LIG.BLUE}";                   <154>
3680 FOR I=1 TO 100:NEXT                      <106>
3690 SYS 64767                                <002>
3700 :                                        <120>
3710 REM +++ SAVE - ROUTINE +++               <200>
3720 :                                        <140>
3730 POKE 198,0:POKE V+21,0:D$=""             <034>
3740 PRINT"{HOME,11DOWN,RIGHT}";              <153>
3750 PRINT"{GREY 3,RVSON,2SPACE}S O U N D{
     2SPACE}S P E I C H E R N{2SPACE,RVOFF
     ,LIG.BLUE}"                              <124>
3760 PRINT"******************************
     **"      "                               <004>
3770 FOR I=0 TO 9                             <028>
3780 PRINT"{RIGHT,32SPACE}"                   <108>
3790 NEXT:PRINT"{9UP}";                       <079>
3800 PRINT"{GREEN,RIGHT}NAME DES SOUNDS ?{
     SPACE,RED}............{12LEFT}";         <065>
3810 GET C$:POKE 204,0                        <238>
3830 IF C$=CHR$(20)THEN D$="":POKE 204,1:P
     RINT"{UP}":GOTO 3800                     <221>
3840 IF C$=CHR$(13)THEN 3940                  <250>
3850 IF C$<"#"OR C$>"Z"THEN 3810              <113>
3855 IF C$="*"OR C$=","OR C$=":"OR C$="?"O
     R C$="@"THEN 3810                        <150>
3860 PRINT C$;:POKE 204,1                     <138>
3870 D$=D$+C$                                 <010>
3880 IF LEN(D$)<11 THEN 3810                  <237>
3890 IF LEN(D$)>11 THEN 3910                  <021>
3900 GOTO 3810                                <206>
3910 GET C$:IF C$=CHR$(20)THEN D$="":POKE
     204,1:PRINT"{UP}":GOTO 3800              <238>
3920 IF C$=CHR$(13) THEN 3950                 <138>
3930 GOTO 3910                                <252>
3940 IF D$=""THEN POKE 204,1:GOTO 410         <137>
3950 PRINT".":POKE 204,1:D$=LEFT$(D$+"....
     ...........",12)                         <249>
3960 UE$="":U$="ABGESPEICHERT......"          <096>
3970 CLOSE 1:OPEN 1,8,2,UE$+"HSY."+D$+",S,
     W":GOSUB 10820                           <227>
3980 IF ER=63 THEN 4030                       <014>
3990 IF ER THEN PRINT"{2DOWN}"                <087>
4000 IF ER AND B$="J"THEN 3970                <251>
4010 IF ER THEN ER=0:CLOSE 1:GOTO 410         <025>
4020 GOTO 4080                                <175>
4030 PRINT"{DOWN,RIGHT,GREY 3}UEBERSCHREIB
     EN{2SPACE}?";                            <217>
4040 POKE 198,0                               <138>
4050 GET A$:IF A$<>"N"AND A$<>"J"AND A$<>C
     HR$(13)THEN 4050                         <135>
4060 IF A$="N"THEN PRINT" NEIN":ER=0:CLOSE
      1:GOTO 410                              <055>
4070 PRINT" JA":UE$="@:":U$="UEBERSCHRIEBE
     N.....":GOTO 3970                        <171>
4080 PRINT"{DOWN,RIGHT,GREEN}SOUND WIRD "U
     $                                        <109>
4090 FOR I=0 TO 2:P1=INT(PU(I)/256):AD(I)=
     A(I)*16+D(I):SR(I)=S(I)*16+R(I)          <030>
4100 PRINT#1,CHR$(WF(I));CHR$(OK(I));CHR$(
     PU(I)-P1*256);CHR$(P1);                  <156>
4110 PRINT#1,CHR$(AD(I));CHR$(SR(I));CHR$(
     FE(I));CHR$(PW(I));CHR$(VI(I));          <069>
4120 NEXT I                                   <140>
4130 C1=INT(CO/256)                           <007>
4140 PRINT#1,CHR$(HP*4+BP*2+TP);CHR$(CO-C1
     *256);CHR$(C1);CHR$(RS);CHR$(VS);        <246>
4150 PRINT#1,CHR$(EF);CHR$(MF);CHR$(MW);CH
     R$(SP);CHR$(RG)                          <167>
4160 CLOSE 1                                  <107>
4170 C$="":IF UE$=""THEN F=F+1:FF$(F)=D$:I
     F F=33 THEN F=32                         <217>
4180 GOTO 410                                 <084>
4190 :                                        <102>
4200 REM +++ LOAD - ROUTINE +++               <229>
4210 :                                        <122>
4220 POKE V+3,65:POKE V+5,107:POKE V+21,6  <070>
4230 A$="={GREY 3,RVSON,SPACE}S O U N D{2S
     PACE}V O N{2SPACE}D I S K{2SPACE}L A
     D E N{SPACE,RVOFF,LIG.BLUE}"             <043>
4235 A1$="={GREY 3,RVSON,5SPACE}S O U N D{
     5SPACE}L A D E N{4SPACE,RVOFF,LIG.BLU
     E}={5SPACE}"                             <118>
4240 GOSUB 10030                              <092>
4250 PRINT"{HOME,13DOWN,RIGHT,GREEN}SOUND{
     SPACE,LIG.RED}"D$"{SPACE,GREEN}WIRD G
     ELADEN.{3UP,LIG.BLUE}"                   <000>
4260 CLOSE 1:OPEN 1,8,2,"HSY."+D$+",S,R":G
     OSUB 10820                               <183>
4270 IF ER AND B$="J"THEN 4260                <233>
4280 IF ER THEN ER=0:CLOSE 1:GOTO 410         <041>
4290 FOR I=0 TO 2                             <165>
4300 GET#1,A$:GOSUB 9730:WF(I)=A%             <137>
4310 GET#1,A$:GOSUB 9730:OK(I)=A%             <148>
4320 GET#1,A$:GOSUB 9730:PU(I)=A%             <041>
4330 GET#1,A$:GOSUB 9730:PU(I)=PU(I)+A%*25
     6                                        <199>
4340 GET#1,A$:GOSUB 9730:AD(I)=A%             <164>
4350 GET#1,A$:GOSUB 9730:SR(I)=A%             <197>
4360 GET#1,A$:GOSUB 9730:FE(I)=A%             <062>
4370 GET#1,A$:GOSUB 9730:PW(I)=A%             <095>
4380 GET#1,A$:GOSUB 9730:VI(I)=A%             <094>
4390 NEXT I                                   <156>
4400 GET#1,A$:GOSUB 9730:Q=A%:HP=(Q AND 4)
     /4:BP=(Q AND 2)/2:TP=Q AND 1             <155>
4410 GET#1,A$:GOSUB 9730:CO=A%                <049>
4420 GET#1,A$:GOSUB 9730:CO=CO+A%*256         <121>
4430 GET#1,A$:GOSUB 9730:RS=A%                <209>
4440 GET#1,A$:GOSUB 9730:VS=A%                <221>
4450 GET#1,A$:GOSUB 9730:EF=A%                <081>
4460 GET#1,A$:GOSUB 9730:MF=A%                <095>
4470 GET#1,A$:GOSUB 9730:MW=A%                <122>
4480 GET#1,A$:GOSUB 9730:SP=A%                <128>
4490 GET#1,A$:GOSUB 9730:RG=A%                <001>
4500 CLOSE 1:CLOSE 15                         <094>
4510 FOR I=0 TO 2                             <131>
4520 RA(I)=(WF(I)AND 128)/128                 <241>
```

Listing 1. »Happysynth« (Fortsetzung)

```
4530 RE(I)=(WF(I)AND 64)/64                     <180>
4540 SZ(I)=(WF(I)AND 32)/32                     <174>
4550 DE(I)=(WF(I)AND 16)/16                     <007>
4560 RI(I)=(WF(I)AND 4)/4                       <021>
4570 SY(I)=(WF(I)AND 2)/2                       <154>
4580 A(I)=(AD(I)AND 240)/16                     <201>
4590 D(I)=AD(I)AND 15                           <214>
4600 S(I)=(SR(I)AND 240)/16                     <083>
4610 R(I)=SR(I)AND 15                           <059>
4620 NEXT I                                     <132>
4630 :                                          <034>
4640 REM +++ SOUND INITIALISIEREN +++           <004>
4650 :                                          <054>
4660 FOR S=0 TO 2                               <107>
4670 POKE WF+3+S,OKT(S)                         <072>
4680 POKE SI+5+S*7,AD(S)                        <080>
4690 POKE SI+6+S*7,SR(S)                        <125>
4700 POKE WF+40+S,PW(S)                         <218>
4710 POKE WF+64+S,VI(S)                         <181>
4720 WF(S)=SY(S)*2+RI(S)*4+DE(S)*16+SZ(S)*
     32+RE(S)*64+RA(S)*128                      <198>
4730 POKE WF+S,WF(S)+8:POKE WF+S,WF(S)          <095>
4740 P1=INT(PU(S)/256)                          <113>
4750 POKE WF+49+S,PU(S)-P1*256:POKE WF+52+
     S,P1                                       <134>
4760 POKE SI+3+S*7,P1:POKE SI+2+S*7,PU(S)-
     P1*256                                     <081>
4770 NEXT S                                     <106>
4780 POKE SI+23,RS*16+FE(0)+FE(1)*2+FE(2)*
     4                                          <231>
4790 POKE SI+24,15+HP*64+BP*32+TP*16            <163>
4800 C1=INT(CO/8):POKE SI+22,C1:POKE SI+21
     ,CO-C1*8:POKE WF+16,C1                     <057>
4810 POKE WF+17,EF                              <117>
4820 POKE WF+6,VS                               <104>
4830 POKE WF+17,EF                              <137>
4840 POKE WF+34,SP                              <214>
4845 POKE WF+19,MF                              <169>
4850 POKE WF+33,MW                              <111>
4860 GOTO 410                                   <000>
4870 :                                          <020>
4880 REM +++ SOUND LOESCHEN +++                 <079>
4890 :                                          <040>
4900 POKE V+3,65:POKE V+5,107:POKE V+21,6       <244>
4910 A$="_{GREY 3,RVSON,5SPACE}S O U N D{4
     SPACE}L O E S C H E N{5SPACE,RVOFF,LI
     G.BLUE}"                                   <092>
4915 A1$="_{GREY 3,RVSON,2SPACE}S O U N D{
     4SPACE}L O E S C H E N{2SPACE,RVOFF,L
     IG.BLUE}_{5SPACE}"                         <245>
4920 GOSUB 10030                                <010>
4930 PRINT"{HOME,13DOWN,RIGHT,GREEN}SOUND{
     SPACE,LIG.RED}"D$"{SPACE,GREEN}LOESCH
     EN ?";                                     <177>
4940 POKE 198,0                                 <022>
4950 GET A$:IF A$="N"THEN 410                   <133>
4960 IF A$<>"J"AND A$<>CHR$(13)THEN 4950        <007>
4970 PRINT"{WHITE,SPACE}JA"                     <001>
4980 CLOSE 1:OPEN 1,8,15,"S:HSY."+D$:GOSUB
     10820                                      <094>
4990 IF ER=1 AND LO=1 THEN FF$(Z1)=FF$(F):
     FF$(F)="":F=F-1:GOTO 5020                  <056>
5000 IF ER=1 THEN PRINT"{DOWN,RIGHT}SOUND
     GESCHUETZT !":PRINT"{DOWN,RIGHT}NOCHM
     AL ?";:GOTO 4950                           <165>
5010 IF B$="J"THEN 4980                         <208>
5020 ER=0:CLOSE 1:GOTO 410                      <253>
7000 ===================================        <221>
7010 ===================================        <231>
7020 === U N T E R P R O G R A M M E ==         <027>
7030 ===================================        <251>
7040 ===================================        <005>
7050 :                                          <168>
7060 REM +++ TITELBILD +++                      <150>
7070 :                                          <188>
7080 IF PEEK(2)=1 OR PEEK(2)=2 THEN RETURN      <060>
7090 POKE 53280,0:POKE 53281,0                  <104>
7100 PRINT"{CLR,CTRL-H,GREEN}";CHR$(142);       <158>
7110 PRINT"********************************
     ********"                                  <116>
7120 PRINT"{ORANGE,RVSON,SPACE}_{SPACE,RIG
     HT}£ ₮{RIGHT,2SPACE}₮{RIGHT,2SPACE}₮{
     RIGHT,SPACE}_{SPACE,RIGHT}£₮₮{RIGHT,S
     PACE}_{SPACE,RIGHT}M{2SPACE,RIGHT,3SP
     ACE,RIGHT,SPACE}_ "                        <079>
7130 PRINT"{RVSON,3SPACE,RIGHT,SPACE}T{SPA
     CE,RIGHT,SPACE}T{RVOFF}£{RVSON,RIGHT,
     SPACE}T{RVOFF}£{RIGHT}₮{RVSON,SPACE,R
     VOFF}£{RIGHT}₮{RVSON,SPACE}₮{RIGHT,RV
     OFF}₮{RVSON,SPACE,RVOFF}£{RVSON,RIGHT
     ,SPACE}M{SPACE,2RIGHT,SPACE,2RIGHT,3S
     PACE}"                                     <200>
7140 PRINT"{RVSON,SPACE}_{SPACE,RIGHT,SPAC
     E}_{SPACE,RIGHT,RVSON,SPACE,3RIGHT,SP
     ACE,4RIGHT,SPACE,2RIGHT,RVOFF}₮{RVSON
     }T{RVOFF}£{2RIGHT,RVSON,SPACE,2RIGHT,
     2SPACE}M{2RIGHT,SPACE,2RIGHT,SPACE}_{
     SPACE,GREEN}"                              <145>
7150 PRINT"********************************
     ********"                                  <156>
7160 PRINT"{RVSON,GREY 2,10SPACE,RVOFF,4SP
     ACE,GREY 2}VERSION 1.0"                    <074>
7170 PRINT"{RVSON,GREY 2}₮₮₮₮₮₮₮₮₮₮₮{RVOFF,
     4SPACE,GREY 2}YYYYYYYYYYY"                 <119>
7180 PRINT"{GREY 2}OWER{RED}B{RVSON,GREY 2
     }₮₮₮₮₮₮"                                   <217>
7190 PRINT"{RVSON,GREY 2}₮₮₮₮₮₮₮₮₮₮₮"SPC(21
     )"{RVOFF,WHITE}F"                          <160>
7200 PRINT"{GREY 2,RVSON,10SPACE}"SPC(20)"
     {RVSON,RED}£ ₮"                            <090>
7210 PRINT"{GREY 2,RVSON}TTTTTTTTTT"SPC(20
     )"{RED,RVSON,3SPACE}"                      <025>
7220 PRINT"{RVSON,GREY 2,10SPACE,GREY 2,RV
     OFF}DEYMB"SPC(15)"{RED}₮{RVSON,SPACE,
     RVOFF}£"                                   <184>
7230 PRINT"{GREY 2,RVSON}**********I{GREY 2
     ,RVOFF,4SPACE}H{GREY 1,RVSON,5SPACE}₮
     "SPC(10)"{GREY 3,SPACE}"                   <027>
7240 PRINT"{GREY 2,RVSON,SPACE}B{RVOFF}UUU
     {RVSON}F{3SPACE}_{GREY 2,RVOFF}FBFNT{
     5SPACE,GREY 1}₮{RVSON}₮"SPC(9)"{GREY
     3,SPACE}"                                  <212>
7250 PRINT"{GREY 2,RVSON,SPACE,RVOFF}JF 1T
     {RVSON,3SPACE}_{RVOFF,LIG.BLUE,5SPACE
     }PORT 2{GREY 1}₮{RVSON}₮{4RIGHT,RED,R
     VOFF}FF{RIGHT,GREY 2,RVSON}£ ₮"            <193>
7260 PRINT"{GREY 2,RVSON,SPACE}C{GREY 1,RV
     OFF}OOO{GREY 2,RVSON}V{3SPACE}_{RVOFF
     ,GREY 1,12SPACE}₮{RVSON}₮"SPC(2)"{BRO
     WN}TTTTTTTTTTT"                            <213>
7270 PRINT"{GREY 2,RVSON,SPACE}B{GREY 1,RV
     OFF}UUU{GREY 2,RVSON}F{3SPACE}_{RVOFF
     ,13SPACE,GREY 1}₮{RVSON,2SPACE,BROWN}
     TTTTTTTTTTT"                               <155>
7280 PRINT"{GREY 2,RVSON,SPACE,RVOFF}JF 3T
     {RVSON,3SPACE}_"SPC(16)"{BROWN,RVOFF}
     ₮{RVSON,9SPACE,RVOFF}£"                    <122>
7290 PRINT"{GREY 2,RVSON,SPACE}C{GREY 1,RV
     OFF}OOO{GREY 2,RVSON}V{3SPACE}_            <014>
7300 PRINT"{GREY 2,RVSON,SPACE}B{RVOFF,GRE
     Y 1}UUU{GREY 2,RVSON}F{3SPACE}_            <200>
7310 PRINT"{GREY 2,RVSON,SPACE,RVOFF}JF 5T
     {RVSON,3SPACE}_{RVOFF,LIG.BLUE,2SPACE
     }t***************************S"            <108>
7320 PRINT"{GREY 2,RVSON,SPACE}C{GREY 1,RV
     OFF}OOO{GREY 2,RVSON}V{3SPACE}_{RVOFF
     ,LIG.BLUE,2SPACE}_{3SPACE,LIG.GREEN}H
     APPYSYNTH{GREY 3,SPACE}(C) 1985{3SPAC
     E,LIG.BLUE}_"                              <061>
7330 PRINT"{GREY 2,RVSON,SPACE}B{RVOFF,GRE
     Y 1}UUU{GREY 2,RVSON}F{3SPACE}_{RVOFF
     ,LIG.BLUE,2SPACE}_{GREY 3}CHRISTIAN Q
     UIRIN SPITZNER{LIG.BLUE}_"                 <227>
7340 PRINT"{GREY 2,RVSON,SPACE,RVOFF}JF 7T
     {RVSON,3SPACE}_{RVOFF,LIG.BLUE,2SPACE
     }_{GREY 3,4SPACE}& BERNHARD CARLI{5SP
     ACE,LIG.BLUE}_"                            <158>
7350 PRINT"{GREY 2,RVSON,SPACE}C{GREY 1,RV
     OFF}OOO{GREY 2,RVSON}V{3SPACE}_{RVOFF
     ,LIG.BLUE,2SPACE}Z*******************
     ******X{UP}"                               <092>
7360 RETURN                                     <050>
7400 :                                          <008>
7410 REM +++ START +++                          <124>
7420 :                                          <028>
7430 IF PEEK(56320)=111 THEN 7470               <241>
7440 PRINT"{HOME,14DOWN}"TAB(27)T$(T1)          <093>
7450 IF T+40<TI THEN T=TI:T1=1-T1               <028>
7460 GOTO 7430                                  <242>
7470 PRINT"{CLR}t**************************
     *************S";                           <252>
7480 FOR I=1064 TO 1944 STEP 40:POKE I,93:
```

```
     POKE I+39,93:PRINT:NEXT                    <131>
7490 PRINT"7********************************
     *********{HOME}";                          <210>
7500 POKE 2023,125                              <009>
7510 RETURN                                     <202>
8000 REM +++ SEITE 1 +++                        <018>
8010 :                                          <112>
8020 PRINT"{HOME,LIG.BLUE}↑***********R****
     ******R***********R*****5";                <093>
8030 PRINT"{LIG.BLUE}={GREEN,RVSON}OSZILLA
     TOR{RVOFF,LIG.BLUE}=HUELLKURVE=TASTAT
     UR{2SPACE}={YELLOW}JOY-{SPACE,LIG.BLU
     E}=";                                      <081>
8040 PRINT"={GREEN,RVSON}& FILTER{2SPACE,R
     VOFF,LIG.BLUE}=MODULATION=DISK-MENUE=
     {YELLOW}STICK{LIG.BLUE}=";                 <197>
8050 PRINT"0***********+**********************+*********
     **+******W";                               <071>
8060 PRINT"={WHITE,2SPACE}DCO 1{SPACE,LIG.
     BLUE,2SPACE}={SPACE,WHITE,SPACE}DCO 2
     {SPACE,LIG.BLUE,2SPACE}={SPACE,WHITE,
     SPACE}DCO 3{SPACE,LIG.BLUE,2SPACE}="       <179>
8070 PRINT"0***********+***********+*********
     **W"                                       <228>
8080 PRINT"=";:FOR I=0 TO 2                     <063>
8090 PRINT F$(DE(I))"NM"F$(SZ(I))" RM"F$(R
     A(I))" TTT{LIG.BLUE,SPACE}=";:NEXT:PR
     INT                                        <200>
8100 PRINT"={10SPACE}={10SPACE}={10SPACE}=
     "                                          <120>
8110 PRINT"=";:FOR I=0 TO 2                     <093>
8120 PRINT F$(RE(I))"OL{SPACE,PURPLE}PW: "
     RIGHT$(STR$(INT(PU(I)/40.95)),2)"%{LI
     G.BLUE}=";:NEXT:PRINT                      <132>
8130 PRINT"0***********+***********+*********
     **W"                                       <032>
8140 PRINT"="F$(RI(0))"RINGMOD. 3{LIG.BLUE
     }=";                                       <217>
8150 PRINT F$(RI(1))"RINGMOD. 1{LIG.BLUE}=
     ";                                         <251>
8160 PRINT F$(RI(2))"RINGMOD. 2{LIG.BLUE}=
     "                                          <177>
8170 PRINT"="F$(SY(0))"SYNCHRO. 3{LIG.BLUE
     }=";                                       <013>
8180 PRINT F$(SY(1))"SYNCHRO. 1{LIG.BLUE}=
     ";                                         <219>
8190 PRINT F$(SY(2))"SYNCHRO. 2{LIG.BLUE}=
     "                                          <145>
8200 PRINT"0***********+***********+*********
     **W"                                       <147>
8210 PRINT"=";:FOR I=0 TO 2                     <072>
8220 PRINT"{PURPLE}OKTAVE: "INT(OKT(I)/12)
     "{LEFT,LIG.BLUE}=";:NEXT:PRINT             <042>
8230 PRINT"=";:FOR I=0 TO 2                     <092>
8240 PRINT"{PURPLE}TONART: "N$(INT(OK(I)-(
     INT(OK(I)/12))*12))"{LIG.BLUE}=";:NEX
     T:PRINT                                    <221>
8250 PRINT"0***********+***********+*********
     **W"                                       <197>
8260 PRINT"={LIG.RED}FILTER "FI$(FE(0))"{L
     IG.BLUE}={LIG.RED}FILTER "FI$(FE(1))"
     {LIG.BLUE}={LIG.RED}FILTER "FI$(FE(2)
     )"{LIG.BLUE}="                             <103>
8270 PRINT"{ORANGE}0*R**********+{LIG.BLUE}*
     **********F***********W"                   <070>
8280 PRINT"{ORANGE}={WHITE}F{ORANGE}={8SPA
     CE,ORANGE}={PURPLE}DCO VERSTIMMUNG:{2
     SPACE}"RIGHT$(STR$(VS),2)"{SPACE,LIG.
     BLUE}="                                    <222>
8290 PRINT"{ORANGE}={WHITE}I{ORANGE}="F$(H
     P)"HOCHPASS{ORANGE}0*****************
     ****W"                                     <135>
8300 PRINT"{ORANGE}={WHITE}L{ORANGE}={8SPA
     CE}={21SPACE}="                            <163>
8310 PRINT"{ORANGE}={WHITE}T{ORANGE}="F$(B
     P)"BANDPASS{ORANGE}=";                     <081>
8320 PRINT"{PURPLE}FREQUENZ: ";RIGHT$("{3S
     PACE}"+STR$(INT(CO*5.8+30)),5);" HERT
     Z{ORANGE}="                                <196>
8330 PRINT"{ORANGE}={WHITE}E{ORANGE}={8SPA
     CE}={21SPACE}="                            <255>
8340 PRINT"{ORANGE}={WHITE}R{ORANGE}="F$(T
     P)"TIEFPASS{ORANGE}={PURPLE}RESONANZ:
      "RIGHT$(STR$(RS),2)"{9SPACE,ORANGE}=
     "                                          <215>
8350 PRINT"7*F*********F*****************
     **{LIG.BLUE}F*****{HOME}"                  <133>
8360 RETURN                                     <197>
8370 :                                          <038>
8380 REM +++ SEITE 2 +++                        <007>
8390 :                                          <058>
8400 PRINT"{HOME,LIG.BLUE}↑***********R****
     ******R***********R*****5";                <173>
8410 PRINT"{LIG.BLUE}=OSZILLATOR={GREEN,RV
     SON}HUELLKURVE{RVOFF,LIG.BLUE}=TASTAT
     UR{2SPACE}={YELLOW}JOY-{SPACE,LIG.BLU
     E}=";                                      <011>
8420 PRINT"=& FILTER{2SPACE}={GREEN,RVSON}
     MODULATION{RVOFF,LIG.BLUE}=DISK-MENUE
     ={YELLOW}STICK{LIG.BLUE}=";                <167>
8430 PRINT"0***********+***********+*********
     **+******W";                               <050>
8440 PRINT"={2SPACE,WHITE}ADSR 1{LIG.BLUE,
     2SPACE}={2SPACE,WHITE}ADSR 2{LIG.BLUE
     ,2SPACE}={2SPACE,WHITE}ADSR 3{LIG.BLU
     E,2SPACE}="                                <131>
8450 PRINT"0*R*********+*R*********+*R*******
     **W"                                       <227>
8460 PRINT"={WHITE}A{LIG.BLUE}=";AD$(A(0))
     ;"={WHITE}A{LIG.BLUE}=";AD$(A(1));"={
     WHITE}A{LIG.BLUE}=";AD$(A(2));"="          <100>
8470 PRINT"0*+*********+*+*********+*+*******
     **W"                                       <037>
8480 PRINT"={WHITE}D{LIG.BLUE}=";R$(D(0));
     "={WHITE}D{LIG.BLUE}=";R$(D(1));"={WH
     ITE}D{LIG.BLUE}=";R$(D(2));"="             <133>
8490 PRINT"0*+*********+*+*********+*+*******
     **W"                                       <057>
8500 PRINT"={WHITE}S{LIG.BLUE}=VOL.: "RIGH
     T$(STR$(S(0)),2)"={WHITE}S{LIG.BLUE}=
     VOL.: "RIGHT$(STR$(S(1)),2);               <183>
8510 PRINT"={WHITE}S{LIG.BLUE}=VOL.: "RIGH
     T$(STR$(S(2)),2)"="                        <073>
8520 PRINT"0*+*********+*+*********+*+*******
     **W"                                       <087>
8530 PRINT"={WHITE}R{LIG.BLUE}=";R$(R(0));
     "={WHITE}R{LIG.BLUE}=";R$(R(1));"={WH
     ITE}R{LIG.BLUE}=";R$(R(2));"="             <163>
8540 PRINT"{ORANGE}0*F*********+*F*********+
     *F*********W"                              <227>
8550 PRINT"=";:FOR I=0 TO 2                     <025>
8560 PRINT"{PURPLE}VIBRATO: ";RIGHT$(STR$(V
     I(I)),2)"{ORANGE}=";:NEXT:PRINT            <120>
8570 PRINT"0***********+***********+*********
     **W"                                       <220>
8580 PRINT"=";:FOR I=0 TO 2                     <055>
8590 PRINT"{PURPLE}PW.MOD.:"RIGHT$(STR$(IN
     T(PW(I)/2.5757)),2)"{ORANGE}=";:NEXT:
     PRINT                                      <208>
8600 PRINT"0***********+***********F*********
     **W"                                       <081>
8610 PRINT"={WHITE}MODULATION{ORANGE}={PUR
     PLE}GESCHWINDIGKEIT:{2SPACE}"RIGHT$("
      "+STR$(SP),3)"{ORANGE}="                  <019>
8620 PRINT"={10SPACE}0*********************
     *W"                                        <070>
8630 PRINT"={10SPACE}0*********************
     *W"                                        <080>
8640 PRINT"={RED}"F$(4+MW)" NM{RED}"F$(5+M
     W)" RM{RED}"F$(3+MW)" OL ";                <205>
8650 PRINT"{ORANGE}={PURPLE}ENVELOPE FOLLO
     W:{2SPACE}"RIGHT$(" "+STR$(EF),3)"{OR
     ANGE}="                                    <058>
8660 PRINT"={10SPACE}={21SPACE}="               <173>
8670 PRINT"={RED}"F$(MW+2)" STIMME 3{SPACE
     ,ORANGE}={PURPLE}FILTER MODULATION:";      <005>
8680 PRINT RIGHT$(" "+STR$(MF),3)"{ORANGE}
     ="                                         <032>
8690 PRINT"7***********F*******************
     **{LIG.BLUE}F{HOME}"                       <095>
8700 RETURN                                     <120>
8710 :                                          <050>
8720 REM +++ SEITE 3 +++                        <248>
8730 :                                          <070>
8740 PRINT"{HOME,LIG.BLUE}↑***********R****
     ******R***********R*****5";                <051>
8750 PRINT"{LIG.BLUE}=OSZILLATOR=HUELLKURV
     E={GREEN,RVSON}TASTATUR{2SPACE,RVOFF,
     LIG.BLUE}={YELLOW}JOY-{SPACE,LIG.BLUE
     }=";                                       <197>
```

Listing 1. »Happysynth« (Fortsetzung)

```
8760 PRINT"=& FILTER{2SPACE}=MODULATION={G
     REEN,RVSON}DISK-MENUE{RVOFF,LIG.BLUE}
     ={YELLOW}STICK{LIG.BLUE}=";              <067>
8770 PRINT"*****************************
     *********";                              <029>
8780 PRINT"={WHITE}SOUND{5SPACE,LIG.BLUE}=
     {WHITE}SOUND{5SPACE,LIG.BLUE}={WHITE}
     SOUND{5SPACE,LIG.BLUE}="                 <146>
8790 PRINT"={WHITE}LOESCHEN{2SPACE,LIG.BLU
     E}={WHITE}SPEICHERN{SPACE,LIG.BLUE}={
     WHITE}LADEN{5SPACE,LIG.BLUE}="           <229>
8800 PRINT"*****************************
     ***"                                     <006>
8810 PRINT"={9SPACE,ORANGE,RVSON,SPACE}{R
     VOFF,2SPACE,RVSON}{2SPACE,RVOFF,SPAC
     E,RVSON,2SPACE}{RVOFF,SPACE,RVSON,SP
     ACE}{RVOFF,9SPACE,LIG.BLUE}="            <149>
8820 PRINT"={9SPACE,ORANGE,RVSON,2SPACE,RV
     OFF}{SPACE,RVSON,SPACE}{SPACE,RVOFF
     ,SPACE,RVSON,SPACE}{SPACE,RVOFF,SPAC
     E,RVSON,2SPACE,RVOFF}{8SPACE,LIG.BLU
     E}="                                     <106>
8830 PRINT"={9SPACE,ORANGE,RVSON,SPACE}{
     RVOFF,SPACE,RVSON,2SPACE}{RVOFF,SPAC
     E,RVSON,2SPACE,RVOFF}£{SPACE,RVSON,SP
     ACE}{RVOFF,8SPACE,LIG.BLUE}="            <099>
8840 PRINT"*****************************
     ***{5SPACE}"                             <243>
8850 PRINT"={32SPACE}={5SPACE}"              <109>
8860 PRINT"={2SPACE,GREY 3,RVSON,SPACE,RVO
     FF,SPACE,RVSON,SPACE,RVOFF,SPACE,RVSO
     N,SPACE}={SPACE,RVOFF,SPACE,RVSON,SPA
     CE,RVOFF,SPACE,RVSON,SPACE,RVOFF,SPAC
     E,RVSON,SPACE}={SPACE,RVOFF,SPACE,RVS
     ON,SPACE,RVOFF,SPACE,RVSON,SPACE}={SP
     ACE,RVOFF,SPACE,RVSON,SPACE,RVOFF,SPA
     CE,RVSON,SPACE,RVOFF}C{RVSON,SPACE,RV
     OFF,LIG.BLUE,3SPACE}={5SPACE}"           <106>
8870 PRINT"{LIG.BLUE}={2SPACE,GREY 3,RVSON
     ,SPACE,RVOFF}2{RVSON,SPACE,RVOFF}3{RV
     SON,SPACE}={SPACE,RVOFF}5{RVSON,SPACE
     ,RVOFF}6{RVSON,SPACE,RVOFF}7{RVSON,SP
     ACE}={SPACE,RVOFF}9{RVSON,SPACE,RVOFF
     }0{RVSON,SPACE}={SPACE,RVOFF}-{RVSON,
     SPACE,RVOFF}£{RVSON,SPACE,RVOFF}L{RVS
     ON,SPACE,RVOFF,LIG.BLUE,3SPACE}="        <185>
8880 PRINT"{LIG.BLUE}={2SPACE,GREY 3,RVSON
     ,SPACE,RVOFF,SPACE,RVSON,SPACE,RVOFF,
     SPACE,RVSON,SPACE}={SPACE,RVOFF,SPACE
     ,RVSON,SPACE,RVOFF,SPACE,RVSON,SPACE,
     RVOFF,SPACE,RVSON,SPACE}={SPACE,RVOFF
     ,SPACE,RVSON,SPACE,RVOFF,SPACE,RVSON,
     SPACE}={SPACE,RVOFF,SPACE,RVSON,SPACE
     ,RVOFF,SPACE,RVSON,SPACE,RVOFF}R{RVSO
     N,SPACE,RVOFF,LIG.BLUE,3SPACE}="         <012>
8890 PRINT"{LIG.BLUE}={2SPACE,GREY 3,RVSON
     ,SPACE}= = = = = = = = = = = = = ={SPAC
     E,RVOFF,LIG.BLUE,3SPACE}="               <164>
8900 PRINT"{LIG.BLUE}={2SPACE,GREY 3,RVSON
     }Q=W=E=R=T=Y=U=I=O=P=@=*=↑=Z{RVOFF,LI
     G.BLUE,3SPACE}="                         <222>
8910 PRINT"={32SPACE}="                      <169>
8920 PRINT"={2SPACE,GREY 3,RVSON,SPACE}={S
     PACE,RVOFF,SPACE,RVSON,SPACE,RVOFF,SP
     ACE,RVSON,SPACE}={SPACE,RVOFF,SPACE,R
     VSON,SPACE,RVOFF,SPACE,RVSON,SPACE,RV
     OFF,SPACE,RVSON,SPACE}={SPACE,RVOFF,S
     PACE,RVSON,SPACE,RVOFF,SPACE,RVSON,SP
     ACE,RVOFF,LIG.BLUE,9SPACE}="             <181>
8930 PRINT"{LIG.BLUE}={2SPACE,GREY 3,RVSON
     ,SPACE}={SPACE,RVOFF}D{RVSON,SPACE,RV
     OFF}F{RVSON,SPACE}={SPACE,RVOFF}H{RVS
     ON,SPACE,RVOFF}J{RVSON,SPACE,RVOFF}K{
     RVSON,SPACE}={SPACE,RVOFF}:{RVSON,SPA
     CE,RVOFF};{RVSON,SPACE,RVOFF,LIG.BLUE
     ,9SPACE}="                               <164>
8940 PRINT"{LIG.BLUE}={2SPACE,GREY 3,RVSON
     ,SPACE}={SPACE,RVOFF,SPACE,RVSON,SPAC
     E,RVOFF,SPACE,RVSON,SPACE}={SPACE,RVO
     FF,SPACE,RVSON,SPACE,RVOFF,SPACE,RVSO
     N,SPACE,RVOFF,SPACE,RVSON,SPACE}={SPA
     CE,RVOFF,SPACE,RVSON,SPACE,RVOFF,SPAC
     E,RVSON,SPACE,RVOFF,LIG.BLUE,9SPACE}=
     "                                        <158>
8950 PRINT"{LIG.BLUE}={2SPACE,GREY 3,RVSON
     ,SPACE}= = = = = = = = = ={SPACE,RVOF
     F,LIG.BLUE,9SPACE}="                     <213>
8960 PRINT"{LIG.BLUE}={2SPACE,GREY 3,RVSON
     }Z=X=C=V=B=N=M=,=.=/={SPACE,RVOFF,LIG
     .BLUE,9SPACE}="                          <041>
8970 PRINT"={32SPACE}="                      <231>
8980 PRINT"*****************************
     ***{HOME}";                              <233>
8990 RETURN                                  <158>
9000 :                                       <086>
9010 REM +++ MPG'S LADEN +++                 <038>
9020 :                                       <106>
9030 IF PEEK(2)=0 THEN POKE 2,1:LOAD"HAPPY
     S.MPG.1",8,1                             <195>
9040 IF PEEK(2)=1 THEN POKE 2,2:LOAD"HAPPY
     S.MPG.2",8,1                             <254>
9050 IF PEEK(2)=2 THEN POKE 2,3:LOAD"HAPPY
     S.MPG.3",8,1                             <057>
9060 RETURN                                  <228>
9100 :                                       <186>
9110 REM +++ INITIALISIEREN +++              <217>
9120 :                                       <206>
9130 OPEN 3,8,15                             <168>
9140 DIM AD$(15),R$(15),N$(11),B(39),FF$(3
     5)                                       <239>
9150 :                                       <236>
9160 FOR I=0 TO 15:READ AD$(I):NEXT          <062>
9170 FOR I=0 TO 15:READ R$(I):NEXT           <031>
9180 FOR I=0 TO 11:READ N$(I):NEXT           <005>
9190 F$(0)=CHR$(28):F$(1)=CHR$(30):FI$(0)=
     "{RED}AUS":FI$(1)="{GREEN}EIN":F$(5)=
     F$(1)                                    <247>
9200 T$(0)="{RED}{LIG.BLUE}":T$(1)="{GRE
     EN}{LIG.BLUE}"                         <199>
9210 :                                       <040>
9220 V=53248:SI=54272:FR=55296:WF=40704:PA
     =1:MW=1                                  <116>
9230 POKE V+32,0:POKE V+33,0:POKE V+21,0:S
     YS 49280                                 <130>
9240 FOR I=0 TO 23:POKE SI+I,0:NEXT:POKE S
     I+24,15                                  <066>
9250 POKE 2041,14:POKE 2042,15:POKE 808,22
     5                                        <172>
9260 POKE 2040,13:POKE V+39,8:POKE V+37,7:
     POKE V+38,0:POKE V+28,1:POKE V+24,21     <036>
9270 POKE V+23,6:POKE V+29,6:POKE V+40,2:P
     OKE V+41,2:POKE 49213,5                  <187>
9280 POKE 49301,203:POKE 49360,203           <000>
9290 FOR I=0 TO 66:POKE WF+I,0:NEXT I:POKE
      WF+33,1:POKE WF+39,26                   <171>
9300 RETURN                                  <214>
9400 :                                       <232>
9410 REM +++ SPRITES +++                     <012>
9420 :                                       <252>
9430 POKE V+21,6:POKE V+40,5:POKE V+41,5:R
     ETURN                                    <186>
9440 POKE V+21,7:POKE V+40,2:POKE V+41,2:R
     ETURN                                    <191>
9450 PRINT"{HOME,8DOWN,LIG.GREEN}"TAB(35)"
     +10{3DOWN,4LEFT}+1{2DOWN,SPACE}-1{3DO
     WN,4LEFT}-10":RETURN                     <093>
9460 PRINT"{HOME,8DOWN}"TAB(34)"{5SPACE,3D
     OWN,5LEFT,2SPACE,2DOWN,3SPACE,3DOWN,5
     LEFT,5SPACE}":RETURN                     <166>
9470 PRINT"{HOME,8DOWN}"TAB(34)"+MSEK{8DOW
     N,5LEFT}-MSEK":RETURN                    <026>
9500 :                                       <078>
9510 REM +++ JOYSTICK-ABFRAGE +++            <064>
9520 :                                       <098>
9530 J=PEEK(56320):IF J AND 15>=16 THEN RE
     TURN                                     <247>
9540 IF J=123 THEN JO=JO+1                   <096>
9550 IF J=119 THEN JO=JO-1                   <109>
9560 IF J=126 THEN JO=JO+10                  <038>
9570 IF J=125 THEN JO=JO-10                  <239>
9580 IF JO<0 THEN JO=0                       <178>
9590 IF JO>255 THEN JO=255                   <015>
9600 BU=J AND 16                             <170>
9610 RETURN                                  <014>
9700 :                                       <022>
9710 REM +++ A$ --> A% +++                   <231>
9720 :                                       <042>
9730 A%=ASC(A$+CHR$(0)):RETURN               <031>
9800 :                                       <124>
9810 REM +++ ROT <-> GRUEN +++               <086>
9820 :                                       <144>
9830 A=PEEK(FR+P):IF A/2=INT(A/2) THEN A=5:
```

```
     GOTO 9850                                <064>
9840 A=2                                      <121>
9850 FOR I=0 TO L:POKE FR+P+I,A:NEXT:RETUR
     N                                        <202>
10000 :                                       <070>
10010 REM +++ LADEN/LOESCHEN +++              <043>
10020 :                                       <090>
10030 PRINT"{HOME,10DOWN}O*******************
      ********************E******"            <060>
10040 PRINT A$                                <056>
10050 POKE 1463,115:POKE 1543,115:POKE V+4
      0,5:POKE V+41,5                         <137>
10060 PRINT"O**************R**************R***
      **********"                             <032>
10070 FOR I=0 TO 10                           <022>
10080 PRINT"_{12SPACE}_{12SPACE}_{12SPACE}
      ":NEXT                                  <132>
10090 PRINT"7**************E**************E***
      *******{HOME}"                          <030>
10100 GOSUB 10210:POKE 1463,93:POKE 1543,9
      3                                       <084>
10110 D$=FF$(Z1):POKE V+3,109:POKE V+5,151
      :POKE V+21,0                            <153>
10120 IF ER THEN ER=0:GOTO 410                <124>
10130 PRINT"{HOME,10DOWN,LIG.BLUE}O********
      ****************************W{5SPACE}"  <203>
10135 PRINT A1$                               <236>
10138 PRINT"{LIG.BLUE}O**********************
      ****************W{5SPACE}"              <114>
10140 FOR I=2 TO 12                           <158>
10150 PRINT"_{32SPACE}_{5SPACE}":NEXT         <042>
10160 PRINT"7**********************************
      ***E{HOME}";                            <141>
10170 RETURN                                  <066>
10180 :                                       <250>
10190 REM +++ SOUND-AUSWAHL +++               <095>
10200 :                                       <014>
10210 IF LL THEN 10590                        <067>
10215 GOSUB 10914:ID$(1)=ID$                  <046>
10220 CLOSE 1:OPEN 1,8,2,"HSY*,S,R":GOSUB
      10820                                   <147>
10225 GOSUB 10914:ID$(1)=ID$                  <056>
10230 IF ER AND B$="J"THEN 10220              <223>
10240 IF ER THEN RETURN                       <100>
10250 A0=0                                    <017>
10260 CLOSE 15:OPEN 15,8,15,"I":GOSUB 1082
      0                                       <124>
10270 IF ER AND B$="J"THEN 10260              <013>
10280 IF ER THEN RETURN                       <140>
10290 F=-1                                    <016>
10300 CLOSE 2:OPEN 2,8,2,"#":GOSUB 10820      <084>
10310 IF ER THEN RETURN                       <170>
10320 IF ER AND B$="J"THEN 10300              <185>
10330 IF ER THEN RETURN                       <190>
10340 TR=18:SE=1                              <067>
10350 PRINT"{HOME,13DOWN,RIGHT}";             <189>
10360 PRINT#15,"U1";2;0;TR;SE                 <159>
10370 GET#2,X$                                <065>
10380 TR=ASC(X$+CHR$(0))                      <067>
10390 GET#2,X$                                <085>
10400 SE=ASC(X$+CHR$(0))                      <235>
10410 FOR X=0 TO 7                            <183>
10420 PRINT#15,"B-P";2;X*32+2                 <102>
10430 GET#2,T$:IF(ASC(T$+CHR$(0))AND 1)=0
      THEN 10560                              <027>
10440 PRINT#15,"B-P";2;X*32+5                 <146>
10450 GET#2,X$:IF X$<>"H"THEN 10560           <092>
10460 GET#2,X$:IF X$<>"S"THEN 10560           <235>
10470 GET#2,X$:IF X$<>"Y"THEN 10560           <248>
10480 GET#2,X$                                <175>
10490 F$=""                                   <123>
10500 FOR Y=0 TO 11                           <073>
10510 GET#2,X$:IF X$=""THEN X$=CHR$(0)        <230>
10520 F$=F$+X$                                <149>
10530 NEXT Y:F=F+1:IF F=33 THEN F=32:GOTO
      10580                                   <230>
10535 FF$(F)=F$                               <254>
10540 PRINT TAB(A0*13+1);"{GREY 2}";F$;       <209>
10550 A0=A0+1:IF A0>2 THEN A0=0:PRINT         <249>
10560 NEXT X                                  <094>
10570 IF TR<>0 THEN 10360                     <086>
10580 CLOSE 15:CLOSE 2:LL=1:GOTO 10620        <093>
10590 GOSUB 10914:IF ID$<>ID$(1)THEN LL=0:
      FOR I=0 TO 32:FF$(I)="":GOTO 10210      <027>
10599 PRINT"{HOME,13DOWN}";                   <225>
10600 A1=-1:FOR I=0 TO 32:A1=A1+1:IF A1>2
```

```
      THEN A1=0:PRINT                         <115>
10610 PRINT TAB(1+A1*13)"{GREY 2}"FF$(I);:
      NEXT I                                  <035>
10620 Z1=0:A1=0:Z2=Z1:A2=A1                   <228>
10630 PRINT"{HOME,11DOWN}"                    <142>
10640 FOR I=0 TO INT(Z2/3):PRINT:NEXT I       <141>
10650 PRINT TAB(A2*13+1)"{GREY 2}"FF$(Z2)     <073>
10660 PRINT"{HOME,11DOWN}"                    <172>
10670 FOR I=0 TO INT(Z1/3):PRINT:NEXT I       <167>
10680 PRINT TAB(A1*13+1)"{WHITE}"FF$(Z1)      <196>
10690 A2=A1:Z2=Z1:J=PEEK(56320)               <181>
10700 IF J=119 THEN Z1=Z1+1:IF Z1>F THEN Z
      1=Z2:GOTO 10720                         <247>
10710 IF J=119 THEN A1=A1+1:IF A1>2 THEN A
      1=0:GOTO 10770                          <203>
10720 IF J=123 THEN Z1=Z1-1:IF Z1<0 THEN Z
      1=Z2:GOTO 10740                         <234>
10730 IF J=123 THEN A1=A1-1:IF A1<0 THEN A
      1=2:GOTO 10770                          <068>
10740 IF J=125 THEN Z1=Z1+3:IF Z1>F THEN Z
      1=Z2:GOTO 10750                         <100>
10750 IF J=126 THEN Z1=Z1-3:IF Z1<0 THEN Z
      1=Z2:GOTO 10770                         <145>
10760 IF J=111 THEN RETURN                    <213>
10770 IF(J AND 15)<15 AND Z1<>Z2 THEN 1063
      0                                       <148>
10780 GOTO 10690                              <078>
10790 :                                       <098>
10800 REM +++ FEHLERMELDUNG +++               <027>
10810 :                                       <118>
10820 INPUT#3,ER,ER$,LO:ER$=ER$+"{16SPACE}
      "                                       <034>
10830 IF ER=0 OR ER=63 OR ER=1 THEN RETURN    <038>
10840 PRINT#3,"UI"                            <200>
10850 FOR I=1 TO 39:B(I)=PEEK(1463+I):NEXT    <142>
10860 PRINT"{HOME,11DOWN,RIGHT,GREY 3}";LE
      FT$(ER$,21);": NOCHMAL ?{6SPACE}"       <138>
10870 IF B(34)=93 THEN PRINT TAB(33)"{UP,L
      IG.BLUE}_"                              <218>
10880 GET B$:IF B$<>"J"AND B$<>"N"THEN 108
      80                                      <090>
10890 IF B$="N"THEN RETURN                    <016>
10900 FOR I=1 TO 39:POKE 1463+I,B(I):NEXT     <124>
10910 RETURN                                  <044>
10911 :                                       <219>
10912 REM +++ ID ABFRAGEN +++                 <039>
10913 :                                       <221>
10914 CLOSE 15:OPEN 15,8,15,"I"               <074>
10915 IF ER AND B$="J"THEN 10914              <028>
10916 IF ER THEN CLOSE 2:CLR 15:RETURN        <254>
10917 CLOSE 2:OPEN 2,8,2,"#"                  <011>
10918 PRINT#15,"U1:";2;0;18;0                 <054>
10919 PRINT#15,"B-P:";2;162                   <038>
10920 GET #2,ID$,I1$:ID$=ID$+I1$              <083>
10930 CLOSE 2:CLOSE 15:RETURN                 <013>
11000 :                                       <052>
11010 REM +++ DATAS +++                       <074>
11020 :                                       <074>
11030 REM +++ DATEN FUER ADSR +++             <232>
11040 DATA"{4SPACE}2 MS","{4SPACE}8 MS","{
      3SPACE}16 MS","{3SPACE}24 MS","{3SPA
      CE}38 MS","{3SPACE}56 MS"               <208>
11050 DATA"{3SPACE}68 MS","{3SPACE}80 MS",
      "{2SPACE}100 MS","{2SPACE}250 MS","{
      2SPACE}500 MS","{2SPACE}800 MS"         <201>
11060 DATA" 1000 MS"," 3000 MS"," 5000 MS"
      ," 8000 MS"                             <222>
11070 DATA"{4SPACE}6 MS","{3SPACE}24 MS","
      {3SPACE}48 MS","{3SPACE}72 MS","{2SP
      ACE}114 MS","{2SPACE}168 MS"            <185>
11080 DATA"{2SPACE}204 MS","{2SPACE}240 MS
      ","{2SPACE}300 MS","{2SPACE}750 MS",
      " 1500 MS"," 2400 MS"                   <120>
11090 DATA" 3000 MS"," 9000 MS","15000 MS"
      ,"24000 MS"                             <060>
11100 REM +++ DATEN FUER NOTEN +++            <113>
11110 DATA" C","#C"," D","#D"," E"," F","#
      F"," G","#G"," A","#A"," H"             <195>
```


Listing 1. »Happysynth« (Schluß)

```
PROGRAMM : HAPPYS.MPG.1  C000 C491
----------------------------------

C000 : EA AC 00 DC 98 29 01 D0  54
C008 : 07 AE 01 D0 CA 8E 01 D0  87
C010 : 98 29 02 D0 07 AE 01 D0  63
C018 : E8 8E 01 D0 98 29 04 D0  26
C020 : 07 AE 00 D0 CA 8E 00 D0  5B
C028 : 98 29 08 D0 07 AE 00 D0  F9
C030 : E8 8E 00 D0 98 29 10 D0  2E
C038 : 03 4C 4D C0 A0 0A A2 00  B2
C040 : CA E0 00 D0 FB 88 C0 00  9B
C048 : D0 F4 4C 01 C0 A2 00 E8  B9
C050 : BD 00 98 CD 00 D0 10 F7  A4
C058 : BD 40 98 CD 00 D0 30 EF  3C
C060 : BD 80 98 CD 01 D0 10 E7  E4
C068 : BD C0 98 CD 01 D0 30 05  C6
C070 : 8E FF 9F EA 60 E0 3E D0  EB
C078 : D6 A2 00 4C 70 C0 00 00  36
C080 : 78 A9 8D 8D 14 03 A9 C0  63
C088 : 8D 15 03 58 60 48 8A 48  6F
C090 : 98 48 08 18 A5 CB C9 40  B2
C098 : D0 1B AD 00 9F 29 FE 8D  BC
C0A0 : 04 D4 AD 01 9F 29 FE 8D  F4
C0A8 : 0B D4 AD 02 9F 29 FE 8D  23
C0B0 : 12 D4 4C 8A C4 AD 00 9F  8A
C0B8 : 09 01 8D 04 D4 AD 01 9F  24
C0C0 : 09 01 8D 0B D4 AD 02 9F  11
C0C8 : 09 01 8D 12 D4 A2 28 A5  46
C0D0 : CB DD 00 C3 F0 06 CA D0  0E
C0D8 : F6 4C 9A C0 18 8A A8 AD  87
C0E0 : 40 9F 8D 84 9F AD 20 9F  0B
C0E8 : 8D 83 9F 20 00 C4 AD 80  01
C0F0 : 9F 48 A9 80 8D 83 9F 20  E1
C0F8 : 00 C4 18 98 6D 03 9F AA  36
C100 : BD 90 C3 8D F0 9F 68 7D  50
C108 : 30 C3 48 AD F0 9F 69 00  93
C110 : 68 38 ED 80 9F 4C E0 C2  85
C118 : EA AD 41 9F 8D 84 9F AD  F4
C120 : 20 9F 8D 83 9F 20 00 C4  68
C128 : AD 80 9F 48 A9 80 8D 83  E2
C130 : 9F 20 00 C4 18 98 6D 04  7C
C138 : 9F AA BD 90 C3 8D F0 9F  59
C140 : 68 7D 30 C3 48 AD F0 9F  E0
C148 : 69 00 8D F0 9F 18 68 6D  6A
C150 : 06 9F 4C 40 C4 EA AD 42  20
C158 : 9F 8D 84 9F AD 20 9F 8D  48
C160 : 83 9F 20 00 C4 AD 80 9F  B6
C168 : 48 A9 80 8D 83 9F 20 00  0C
C170 : C4 18 98 6D 05 9F AA BD  88
C178 : 90 C3 8D F0 9F 68 7D 30  FF
C180 : C3 48 AD F0 9F 69 00 8D  51
C188 : F0 9F 68 4C 60 C4 EA EA  99
C190 : EA EA EA AD 1C D4 8D 84  07
C198 : 9F AD 11 9F 8D 83 9F 20  FA
C1A0 : 00 C4 AD 81 9F 48 AD 80  92
C1A8 : 9F 48 AD 20 9F 8D 84 9F  92
C1B0 : AD 13 9F 8D 83 9F 20 00  36
C1B8 : C4 68 6D 0F 9F 6D 80 9F  94
C1C0 : 08 4A 4A 4A 4A 4A 8D 15  21
C1C8 : D4 28 68 6D 81 9F 6D 10  63
C1D0 : 9F 90 02 A9 FF 8D 16 D4  DB
C1D8 : EA EA EA AD 22 9F 29 FC  65
C1E0 : D0 14 AD 23 9F 6D 22 9F  B7
C1E8 : 8D 23 9F C9 04 30 2E A9  F6
C1F0 : 00 8D 23 9F A9 04 4A 4A  EC
C1F8 : 8D 26 9F AD 24 9F 38 ED  32
C200 : 26 9F 8D 24 9F B0 08 AD  D9
C208 : 25 9F 49 FF 8D 25 9F 18  00
C210 : AD 21 9F C9 00 D0 09 AD  75
C218 : 24 9F 8D 20 9F 4C 4C C2  86
C220 : C9 01 D0 15 AD 25 9F C9  57
C228 : 00 F0 08 38 ED 24 9F 18  58
C230 : 4C 1A C2 AD 24 9F 4C 1A  94
C238 : C2 C9 02 D0 06 AD 25 9F  1B
C240 : 4C 1A C2 AD 1B D4 4C 1A  BD
C248 : C2 EA EA EA AE 27 9F AD  95
C250 : 20 9F 9D 00 D4 A0 00 AD  55
C258 : 20 9F 8D 84 9F B9 28 9F  E3
C260 : 8D 83 9F 20 00 C4 AD 80  79
C268 : 9F 4A 4A 4A 4A 99 2B 9F  66
C270 : 18 AD 81 9F 0A 0A 0A 0A  E0
C278 : 18 79 2B 9F 99 2B 9F 08  8D
C280 : AD 81 9F 4A 4A 4A 4A 28  8F
C288 : 69 00 99 2E 9F C8 C0 03  67
C290 : D0 C5 EA EA EA A2 00 A0  60
C298 : 00 20 B1 C2 E8 A0 07 20  5D
C2A0 : B1 C2 E8 A0 0E 20 B1 C2  2F
C2A8 : 28 68 A8 68 AA 68 4C 31  BD
C2B0 : EA BD 28 9F D0 10 BD 31  5E
C2B8 : 9F 99 02 D4 BD 34 9F 99  6E
C2C0 : 03 D4 60 00 00 00 18 BD  21
C2C8 : 31 9F 7D 2B 9F 99 02 D4  06
C2D0 : BD 34 9F 7D 2E 9F C9 10  66
C2D8 : 30 02 A9 0F 99 03 D4 60  1B
C2E0 : 8D 00 D4 AD F0 9F B0 03  2D
C2E8 : 38 E9 01 8D 01 D4 4C 19  21
C2F0 : C1 00 00 00 00 00 00 00  B2
C2F8 : 00 00 00 00 00 00 00 00  F9
C300 : 00 3E 3B 09 08 0E 11 10  65
C308 : 16 13 19 18 1E 21 20 26  A9
C310 : 23 29 2E 2B 31 30 36 33  8D
C318 : 0C 17 12 14 15 1F 1C 1D  AC
C320 : 27 22 24 25 2F 2C 2D 37  7E
C328 : 32 00 00 00 00 00 00 00  5B
C330 : 16 27 39 4B 5F 74 8A A1  99
C338 : BA D4 F0 0E 2D 4E 71 96  92
C340 : BE E7 14 42 74 A9 E0 1B  8D
C348 : 5A 9C E2 2D 7B CF 27 85  2C
C350 : E8 51 C1 37 B4 38 C4 59  0B
C358 : F7 9E 4E 0A D0 A2 81 6D  76
C360 : 67 70 89 B2 ED 3B 9D 14  0F
C368 : A0 45 03 DB CF E1 12 65  06
C370 : DB 76 3A 27 41 8A 05 B5  E2
C378 : 9D C1 24 C9 B6 ED 73 4E  7D
C380 : 82 14 0A 6A 3B 82 48 93  EC
C388 : 6B DA E7 9C 04 28 14 00  C0
C390 : 01 01 01 01 01 01 01 01  90
C398 : 01 01 01 02 02 02 02 02  D7
C3A0 : 02 02 03 03 03 03 03 04  21
C3A8 : 04 04 04 05 05 05 06 06  ED
C3B0 : 06 07 07 08 08 09 09 0A  FE
C3B8 : 0A 0B 0C 0D 0D 0E 0F 10  8A
C3C0 : 11 12 13 14 15 17 18 1A  C0
C3C8 : 1B 1D 1F 20 22 24 27 29  70
C3D0 : 2B 2E 31 34 37 3A 3E 41  A6
C3D8 : 45 49 4E 52 57 5C 62 68  52
C3E0 : 6E 75 7C 83 8B 93 9C A5  AB
C3E8 : AF B9 C4 D0 DD EA F8 00  D8
C3F0 : 00 00 00 00 00 00 00 00  F1
C3F8 : 00 00 00 00 00 00 00 00  F9
C400 : AD 83 9F 48 AD 84 9F 48  6E
C408 : A9 00 8D 80 9F 8D 81 9F  D0
C410 : 8D 82 9F A2 08 4E 83 9F  5B
C418 : 90 13 AD 80 9F 18 6D 84  27
C420 : 9F 8D 80 9F AD 81 9F 6D  DA
C428 : 82 9F 8D 81 9F 0E 84 9F  C9
C430 : 2E 82 9F CA D0 DF 68 8D  A9
C438 : 84 9F 68 8D 83 9F 60 00  0E
C440 : 48 AD F0 9F 69 00 8D F0  3E
C448 : 9F 38 68 ED 80 9F 8D 07  24
C450 : D4 AD F0 9F B0 03 38 E9  03
C458 : 01 8D 08 D4 4C 56 C1 00  3B
C460 : 38 ED 06 9F 48 AD F0 9F  F9
C468 : B0 03 38 E9 01 8D F0 9F  65
C470 : 68 38 ED 80 9F 8D 0E D4  C8
C478 : AD F0 9F B0 03 38 E9 01  37
C480 : 8D 0F D4 8C F1 9F 4C 93  D0
C488 : C1 00 AE F1 9F 4C DC C0  84
C490 : FF 00 FF 00 FF 00 FF 00  8F
```

Listing 2. Maschinenprogramm 1. Wird vom Hauptprogramm nachgeladen. Zur Eingabe bitte den MSE (Seite 18) verwenden.

```
PROGRAMM : HAPPYS.MPG.2      9800 9901
--------------------------------------

9800 : 00 00 66 BE 00 66 BE 00  A0
9808 : 28 3E 66 80 96 BE D9 EE  9E
9810 : 00 80 00 28 66 80 BE D9  6F
9818 : 00 80 00 66 BE 00 66 BE  28
9820 : 00 66 BE 00 66 BE 00 66  2C
9828 : BE 00 66 BE 00 66 BE 00  86
9830 : 66 BE 00 30 46 66 BE 00  8E
9838 : 66 BE 00 66 BE 00 80 00  B8
9840 : 00 66 BE FF 66 BE FF 28  CF
9848 : 3E 66 80 96 BE D9 EE FF  23
9850 : 80 FF 28 66 80 BE D9 FF  0C
9858 : 80 FF 66 BE FF 66 BE FF  77
9860 : 66 BE FF 66 BE FF 66 BE  F5
9868 : FF 66 BE FF 66 BE FF 66  73
9870 : BE FF 30 46 66 BE FF 66  2C
9878 : BE FF 66 BE FF 80 FF 00  AB
9880 : 00 00 00 00 4D 4D 4D 5F  B4
9888 : 5F 5F 5F 5F 5F 5F 5F 5F  88
9890 : 69 69 71 71 71 71 71 71  84
9898 : 7B 7B 80 80 80 89 89 89  8E
98A0 : 92 92 92 99 99 99 A1 A1  83
98A8 : A1 AD AD AD BD BD BD D2  A7
98B0 : D2 D2 D9 D9 D9 D9 D9 E1  34
98B8 : E1 E1 E9 E9 E9 F3 F3 00  4F
98C0 : 00 4D 4D 4D 5F 5F 5F 69  A5
98C8 : 69 69 69 69 69 69 69 69  C8
98D0 : 71 71 7B 7B 7B 7B 7B 7B  C1
98D8 : 80 80 89 89 89 92 92 92  C8
98E0 : 99 99 99 A1 A1 A1 AD AD  1A
98E8 : AD BD BD BD D2 D2 D2 D9  5E
98F0 : D9 D9 E1 E1 E1 E1 E1 E9  F3
98F8 : E9 E9 F3 F3 F3 FF FF 00  90
9900 : 00 01 FF 31 00 02 FF 00  B7
```

Listing 3. Maschinenprogramm 2. Wird vom Hauptprogramm nachgeladen. Zur Eingabe bitte den MSE (Seite 18) verwenden.

```
PROGRAMM : HAPPYS.MPG.3      033E 03F
--------------------------------------

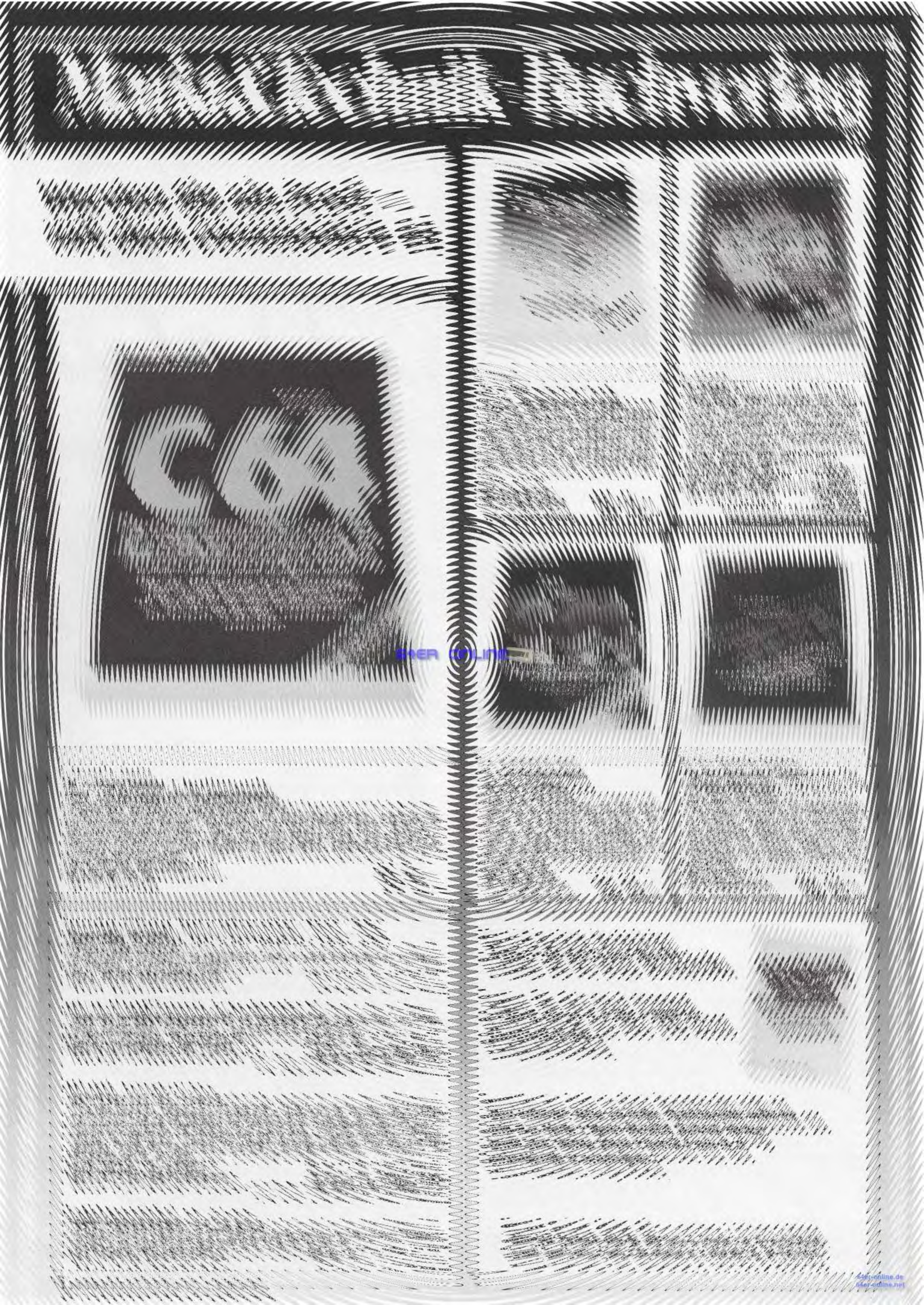
033E : 00 00 00 38 00 00 3A 00  2E
0346 : 00 3A 00 00 3A 00 00 3A  7C
034E : 00 00 3A E0 00 3A EE 00  87
0356 : 3A EE 38 3A AE 0E 3A AA  F6
035E : 0E BA AA 03 AA AA 03 AA  36
0366 : AA 00 EA AA 00 3A A8 00  95
036E : 3A A8 00 FA AB 00 FE AD  6E
0376 : 00 D7 F5 00 D5 55 00 D5  93
037E : 55 00 00 00 00 00 00 00  D4
0386 : 00 00 00 00 00 00 00 00  87
038E : 00 00 00 00 00 00 00 00  8F
0396 : 10 00 00 38 00 00 7C 00  9F
039E : 00 10 00 00 10 00 00 10  C8
03A6 : 00 00 10 00 00 10 00 00  2B
03AE : 10 00 00 10 00 00 FE 00  BC
03B6 : 10 FE 08 30 FE 0C 7F FF  9C
03BE : FE 00 30 FE 0C 10 FE 08  F6
03C6 : 00 FE 00 00 10 00 00 10  67
03CE : 00 00 10 00 00 10 00 00  53
03D6 : 10 00 00 10 00 00 10 00  29
03DE : 00 7C 00 00 38 00 00 10  C0
03E6 : 00 00 00 00 00 00 00 00  E7
03EE : 00 00 00 00 00 00 00 00  EF
03F6 : 00 00 00 00 00 00 00 00  F7
03FE : FF 00 20 10 12 0F 07 12  E1
```

Listing 4. Maschinenprogramm 3. Wird vom Hauptprogramm nachgeladen. Zur Eingabe bitte den MSE (Seite 18) verwenden.

```
100 REM +++ DEMO-SOUNDS +++                          <255>
110 FOR I=1 TO 6                                     <083>
120 :  READ NA$                                      <177>
130 :  OPEN 1,8,2,NA$+",S,W"                         <061>
140 :  FOR T=1 TO 37                                 <090>
150 :    READ S                                      <239>
160 :    PRINT#1,CHR$(S);                            <122>
170 :  NEXT T                                        <235>
180 :  CLOSE 1                                       <180>
190 NEXT I                                           <018>
200 END                                              <202>
210 DATA "HSY.ORGEL......."                          <134>
220 DATA 32,28,,4,,107,1,1,,32,24,,4,                <030>
230 DATA 107,1,1,,32,31,,4,,107,1,1,,1               <057>
240 DATA 168,,5,1,87,,1,14,25                        <145>
250 DATA "HSY.BRANDUNG...."                          <202>
260 DATA 128,28,,4,220,255,1,1,,128,24               <151>
270 DATA ,4,220,255,1,1,,128,31,,4,220               <053>
280 DATA 255,1,1,,1,168,,15,5,87,80,1                <226>
290 DATA 10,25                                       <105>
300 DATA "HSY.FLOETE......"                          <081>
310 DATA 16,43,,,11,23,,,,16,43,16,8                 <051>
320 DATA 11,240,,,,16,43,,,11,133,,,,                <070>
330 DATA ,,,30,,,1,,                                 <234>
340 DATA "HSY.GLOCKEN....."                          <225>
350 DATA 20,31,,,10,108,,,,20,16,,,10                <001>
360 DATA 108,,,,16,39,,,10,108,,,,,,,                <113>
370 DATA ,,,1,,                                      <184>
380 DATA "HSY.BASS........"                          <193>
390 DATA 34,19,,,12,250,1,,,32,31,16,8               <250>
400 DATA 12,250,1,,,32,7,,,10,56,1,,,1               <073>
410 DATA 64,1,14,4,155,,2,1,                         <131>
420 DATA "HSY.METALLIC...."                          <103>
430 DATA 20,43,,,10,56,,,,20,12,,,10                 <063>
440 DATA 56,,,,20,43,,,10,56,,,,,207,2,              <220>
450 DATA 2,130,,1,,                                  <154>

© 64'er
```

Listing 5. Demosounds zu »Happysynth«. Nach Starten mit »RUN« werden die Sounds automatisch auf Diskette gespeichert. Zur Eingabe bitte den Checksummer (auf Seite 16) verwenden.



HI-EDDI: ein fantastisches Zeichen- und Malprogramm

HI-EDDI ist ein High-Resolution-Grafik-Editor, der vieles bietet, was man bei teuren, kommerziellen Programmen vergeblich sucht. Sehen Sie sich die Bilder an und Sie bekommen einen Eindruck von seinen Fähigkeiten.

Hier eine Aufstellung der herausragenden Eigenschaften:

- Wahlweise als »Schwarzweiß«-Zeichenprogramm oder als farbiges Malprogramm verwendbar, in jedem Fall mit der vollen Auflösung von 320 x 200 Bildpunkten.
- 7 (in Worten: sieben) Schwarzweiß-Bildschirme oder 6 farbige stehen zur Verfügung.
- Integrierter, leistungsfähiger Sprite-Editor (mit Sprite spiegeln, drehen...) und die Fähigkeit, Sprites aus dem Grafikbild herauszukopieren oder einzupflanzen. Damit ist zum Beispiel Ausschnittvergrößerung (Zoom) ebenso wie das Erstellen von »Construction Sets« möglich.
- Mit einer Zeichentrickfilm-Funktion sind fast flimmerfreie Bewegungsabläufe mit bis zu 24 Bildern pro Sekunde möglich.
- Befehlseingabe wahlweise über Tastatur oder eine selbst zu gestaltende Menütafel (à la Koalapainter).
- Zusammenhängender Ausdruck mehrerer Bilder möglich, das ergibt Superbilder mit einer Breite von minimal 640 Punkten und unbegrenzter Länge (Bild 2).
- Ferner sind alle »Standardbefehle« wie Draw, Line, Rectangle, Circle, Paint, Move und Text vorhanden.

Natürlich ist klar, daß sowas nur in Maschinensprache geht, und die ist mit rund 4700 DATAs nicht gerade kurz. Gemessen an den Möglichkeiten des Programms ist es allerdings auch nicht viel. Die Bilder auf diesen Seiten lassen die Vielfältigkeit von HI-EDDI nur ahnen. Dabei lassen sich nicht nur mit HI-EDDI erzeugte Bilder laden und bearbeiten, sondern auch Grafikbilder anderer Programme, wie von Koala oder Paint Magic, zum Beispiel die der Dia-Show.

Nach dem Start meldet sich HI-EDDI mit der Frage »Betriebsart«. Wird hier 0 eingegeben (oder einfach RETURN), wird HI-EDDI als »Schwarzweiß«-Programm betrieben, bei 128 dagegen als farbiges Malprogramm. Weitere mögliche Eingaben werden später behandelt, ebenso die Befehlseingabe mittels Menütafel. Zunächst werden alle Befehle, die im folgenden zusammengestellt sind, durch Tastendruck eingegeben.

D Draw — »Freihändig« zeichnen

Im Draw-Modus kann mit dem kreuzförmigen Cursor (der wird mit einem Joystick in Port 2 gesteuert) »freihändig« gezeichnet werden. Bei gedrücktem Feuerknopf werden die überfahrenen Punkte gesetzt. Um Punkte zu löschen, muß zusätzlich die SHIFT-Taste (oder SHIFT-LOCK zum Feststellen) gedrückt werden.

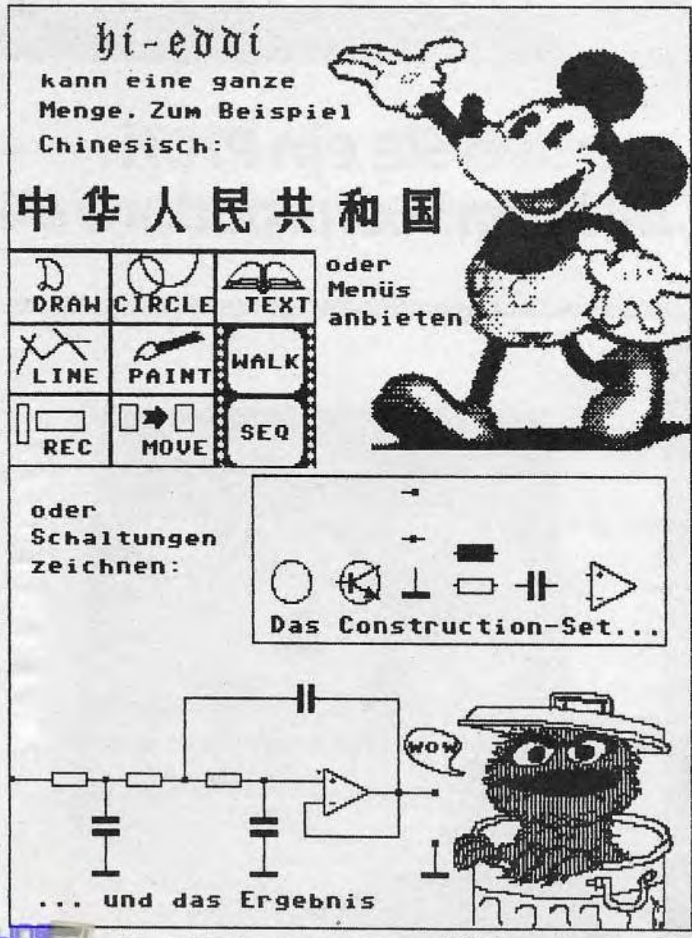


Bild 1. In dieser Hardcopy wurden zwei Bilder untereinander ausgedruckt. Dies geschieht nahtlos. Es können zwei Bilder nebeneinander und unbegrenzt viele untereinander gesetzt werden. Siehe dazu auch Bild 2.

L Line — Linien ziehen

Mit dem ersten Knopfdruck am Joystick wird der Anfangspunkt einer Linie festgelegt, mit dem zweiten der Endpunkt, mit dem dritten wieder ein Anfangspunkt etc.. Der jeweilige Anfangspunkt wird auf F7 gespeichert (siehe Funktionstasten), damit lassen sich Strahlen besonders einfach zeichnen. SHIFT wie bei D.

R Rectangle — Rechtecke zeichnen

Wie L — auch bezüglich SHIFT und F7 —, jedoch wird zwischen den zwei markierten Punkten ein Rechteck gezeichnet.

C Circle — Kreise zeichnen

Der erste Knopfdruck ergibt den Mittelpunkt (auf F7 gespeichert, damit ist einfaches Zeichnen konzentrischer Kreise möglich), der zweite einen beliebigen Randpunkt, von dem aus HI-EDDI im Uhrzeigersinn einen Kreis zieht. Wird beim Anstoßen an den Rand abgebrochen.

Möchte man einen Ausschnitt eines Kreises, der nicht ganz auf den Bildschirm paßt, zeichnen, so muß der zweite Knopfdruck einen Punkt am Bildschirmrand markieren, von dem aus der sichtbare Kreisausschnitt im Uhrzeigersinn gezeichnet werden kann. Allerdings darf der Radius maximal 256 Punkte betragen.

P Paint — Ausmalen begrenzter Flächen

Cursor mitten auf die auszumalende Fläche setzen und Knopf drücken. Sollte durch ein Loch in der Umrandung der ganze Bildschirm vollzulaufen drohen, so kann der Vorgang durch nochmaligen Knopfdruck abgebrochen werden (dazu ist allerdings eine gute Reaktion nötig, denn HI-EDDI füllt Flächen um einiges schneller als Simons Basic). Zum Löschen von Flächen: Bildschirm mit »I« invertieren, entstandenes »Loch« vollaufen lassen, zurückinvertieren.

M Move — Verschieben von Bildschirmbereichen

Mit den ersten beiden Knopfdrücken werden zwei diagonale Ecken des zu transportierenden Bereiches gesetzt (nach dem zweiten Knopfdruck erscheint eine farbliche Markierung dieses Bereiches), der dritte Knopfdruck gibt die linke, obere Ecke des Zielbereiches an. Der Zielbereich muß noch ganz auf den Bildschirm passen (sonst wird der Knopfdruck nicht akzeptiert), er darf jedoch den Quellbereich überlappen oder in einem anderen Bildschirm liegen. Die Auflösung des Move-Befehls entspricht der des normalen Textbildschirmes, es kann also nur im 40 x 25-Raster verschoben werden.

Soll ein Bereich mehrmals kopiert werden, so muß er nicht jedesmal neu markiert werden: Ein Druck auf die Pfeil-nach-oben-Taste holt die letzte Markierung wieder auf den Bildschirm, es kann danach sofort der Zielbereich bestimmt werden (funktioniert nur, solange zwischenzeitlich kein Moduswechsel erfolgte).

Hat man sich beim Markieren vertan, so löscht ein Druck auf die Pfeil-nach-links-Taste die Markierung wieder (gilt auch für L, R, C).

T Text — Buchstaben und Grafiksymbole einfügen

Es erscheint ein 8 x 8-Pixel großer Rahmen, der sich fast so benimmt, wie der Blinkcursor im Textbildschirm: Drucken von Buchstaben und Grafikzeichen, Cursorsteuerung mittels Cursortasten, Löschen mittels DEL (rückwärts) und INST (vorwärts, um für Text Platz zu schaffen), Reverse on/off und Umschaltung der beiden Zeichensätze mit C = SHIFT (es können alle 512 Zeichen der beiden Zeichensätze gleichzeitig dargestellt werden!).

Daneben bleibt die Cursorsteuerung mittels Joystick erhalten, ebenso alle anderen Befehle, die jedoch nur durch gleichzeitiges Drücken der CTRL-Taste eingegeben werden können. Durch Anwahl eines anderen Modus, zum Beispiel CTRL D für Draw, wird der Textmodus verlassen und es ist wieder »Ein-Hand-Eingabe« der Befehle möglich.

G Get Sprite — Sprite aus Bildschirm kopieren

In diesem und den folgenden drei Modi erscheint ein spritegroßer Rahmen als Cursor. Auf Knopfdruck wird der Bildschirmausschnitt, auf dem der Rahmen sitzt, in das Sprite hineinkopiert, anschließend geht HI-EDDI automatisch in den Append-Modus, das Sprite kann an anderer Stelle wieder eingepflanzt oder im Sprite-Editor bearbeitet werden.

A Append — Sprite in Bildschirm einfügen

Auf Knopfdruck wird der Spriteinhalt in den Bildschirm eingefügt, ohne jedoch den Bildschirmausschnitt vorher zu löschen (Oder-Verknüpfung). Bei gleichzeitiger Bewegung wird das Sprite zum »programmierbaren Pinsel«.

S Stamp — Sprite auf Bildschirm kleben

Wie A, jedoch wird vor dem Einfügen der Untergrund gelöscht, das Sprite wird wie eine Briefmarke auf den Bildschirm geklebt.

E Erase — Löschen

Der Rahmen wird zum Radiergummi, der alles löscht, was er überfährt.

F Foreground-Colourmode — Vordergrund einfärben
B Background-Colourmode — Hintergrund einfärben

Diese beiden Modus-Befehle sind nur wirksam, wenn HI-EDDI als farbiges Malprogramm betrieben wird. Doch dazu ist vorweg einiges zum Konzept zu sagen:

Im Gegensatz zu den meisten käuflichen Programmen, die im Multicolour-Modus arbeiten (mehrere Farben, aber nur halbe Auflösung), ist HI-EDDI konsequent als Zeichenprogramm mit maximaler Auflösung konzipiert. Die Farbfähigkeiten sind nur ein »Nebenprodukt« und deshalb nicht so ausgeprägt: Pro 8 x 8-Punkte-Feld (entsprechend einer Position im Textbildschirm) stehen nur zwei Farben — je eine für Vorder- und Hintergrund — zur Verfügung. Bei insgesamt 1 000 Feldern und 16 Farben kann das allerdings auch recht bunt werden. Außerdem ist es durch die Trennung von Zeichnen und Einfärben möglich, bestehende »Schwarzweiß-Bilder« leicht nachträglich zu colorieren.

Nun zu den Befehlen **F** und **B**: Sie schalten einerseits die Rahmenfarbe (und die Farbe des Sprite-Inhalts bei **A** und **S**) weiter, andererseits wählen sie den Fore- beziehungsweise Back-Mode an, in dem auf Knopfdruck die gesetzten Pixel (=Vordergrund) beziehungsweise gelöschten Pixel (= Hintergrund) des 8 x 8-Feldes, auf dem sich der Cursor befindet, mit der Rahmenfarbe eingefärbt werden. Beispiel: Um den Vordergrund gelb anzumalen, muß die Taste F so oft gedrückt werden, bis der Rahmen gelb ist. Dann kann mit dem Cursor gepinselt werden.

Hat man ein Feld zuviel angepinselt, so kann durch gleichzeitiges Drücken der SHIFT-Taste (wie bei D, L, R, C) die gerade gemalte Farbe gelöscht werden. Genauer gesagt: Es erscheint wieder die Farbe, die beim letzten Bildschirmwechsel dort war. Als Bildschirmwechsel gelten:

1. Bildschirmspeicherwechsel (1 bis 7)
2. Ausschalten des High-Resolution-Bildschirmes (Sprite-Editor, Befehle H, V, SHIFT W, Disk- und Druckerbefehle).
3. Move-Befehl, auch wenn nur innerhalb eines Bildes »gemovet« wurde.

Die Pfeil-nach-links-Taste (Korrekturtaste) hat die gleiche Wirkung wie SHIFT + Knopfdruck, jedoch für den ganzen Bildschirm: Es werden alle Farbveränderungen seit dem letzten Bildschirmwechsel rückgängig gemacht.

Direkte Befehle

Alle bisher beschriebenen Befehle haben eins gemeinsam: Sie dienen zur Anwahl eines Modus, sie haben also eine »Nachwirkung«, vor allem, was die Funktion des Feuerknopfes anbelangt. Die nun folgenden Befehle ändern den gerade eingestellten Modus nicht, sie haben nur eine unmittelbare Wirkung.

1 bis 7 beziehungsweise 1 bis 6: Bildschirmspeicher-Anwahl

HI-EDDI hat im Schwarzweiß-Betrieb sieben und im Farbbetrieb sechs Bildschirmspeicher, die durch Eingabe ihrer Nummer auf den Bildschirm geholt werden.

Wozu so viele? Beispiele: Abspeichern von verschiedenen Zwischenstadien der bearbeiteten Werke, bei denen man wieder ansetzen kann, wenn man was vermurkst hat. Oder Erstellen von »Construction Sets«: Aus einer Anzahl zum Beispiel von Schaltsymbolen kann man mittels der Sprite-Befehle im Nu Schaltpläne aufbauen. Für die Befehle **Print** und **Walk** sind sieben Speicher eigentlich zu wenig, aber im C 64 haben eben nicht mehr Platz!

I Invertieren

Die Pixel des Bildschirmes werden invertiert. In Zusammenhang mit den Befehlen Und, Or, Exor, Append, Stamp, Get, Erase, Paint und Print ergibt das eine Vielzahl von Möglichkeiten.

Un Und-Verknüpfung

Der aktuelle (= sichtbare) Bildschirm und der Bildschirm Nummer n werden Und-verknüpft, das Ergebnis im aktuellen Bildschirm abgelegt.

On Oder-Verknüpfung

Wie U, jedoch Oder-Verknüpfung. Eignet sich zum Beispiel zum Duplizieren von Bildern. Da die Befehle I, U, O, X keinen Einfluß auf die Farbinformation eines Bildes haben, wird sie bei Farb-Betrieb nicht dupliziert. Soll dies geschehen, so muß mit dem Move-Befehl dupliziert werden.

Xn Exor-(Exclusiv-Oder-)Verknüpfung

Zweimalige Exor-Verknüpfung mit demselben Bild bewirkt, daß dieses wieder aus dem sichtbaren Bild »herausgefieselt« wird.

F, B (Fore, Back) — Rahmenfarbe weiterschalten
SHIFT F Total Foreground — Vordergrund-Farbe
SHIFT B Total Background — Hintergrund-Farbe

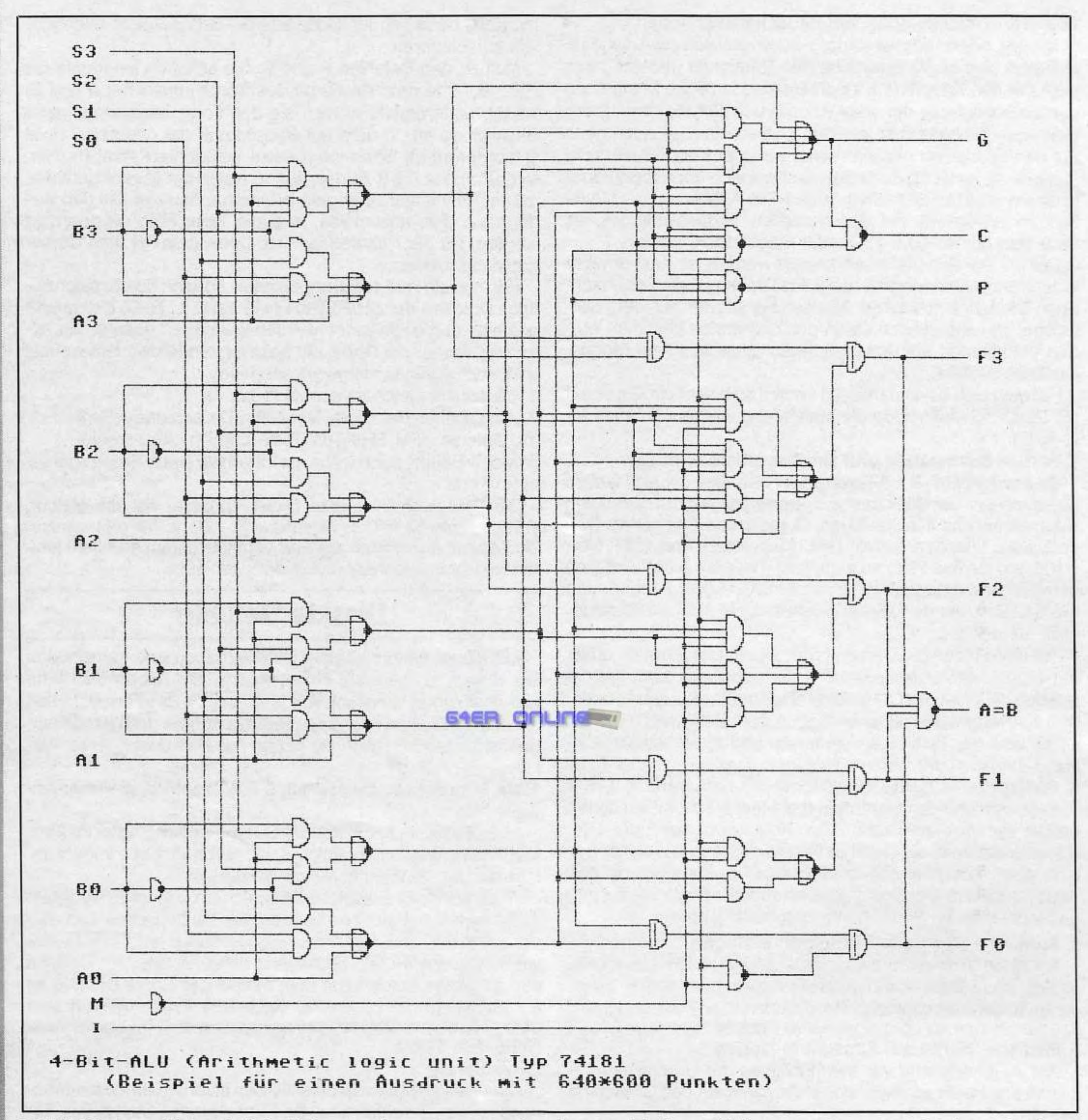


Bild 2. Dieser verkleinerte Schaltplan wurde aus insgesamt sechs verschiedenen Bildern zusammengesetzt. Je zwei nebeneinander und drei untereinander. Auch ein reverser Ausdruck ist mit einem Tastendruck möglich. Die Symbole aus dem Construction-Set können einzeln in jedes beliebige Bild hineinkopiert werden.

Die Befehle F und B wurden bereits bei den Modusbefehlen behandelt. Im Schwarzweiß-Betrieb schalten diese Befehle nur die Rahmenfarbe weiter, den aktuellen Modus verändern sie nicht.

SHIFT F und SHIFT B sind dagegen in beiden Betriebsarten (farbig und schwarzweiß) gleich: Sie färben den gesamten Vorder- oder Hintergrund mit der aktuellen Rahmenfarbe ein. Sollte man auf diese Art im Farb-Betrieb versehentlich eine mühsam erstellte Colorierung löschen: Pfeil-nach-links-Taste drücken, und sie ist wieder da.

SHIFT CLR Bildschirm löschen

Die Farbinformation wird nicht gelöscht und kann somit isoliert werden.

+ Schnelle Cursorgeschwindigkeit (beschleunigend)

— Langsame Cursorgeschwindigkeit

HI-EDDI besitzt einen beschleunigenden Cursor: Er ist langsam genug, um durch Antippen des Joysticks pixelweise zu rangieren, bei größeren Entfernungen wird er jedoch — ohne lästiges Umschalten — schneller. Da jedoch die Beschleunigung manchmal unerwünscht ist, läßt sie sich durch Drücken der Minus-Taste abschalten.

F1 bis F8: Tabulatoren

Die vier Funktionstasten dienen als Speicher für vier Cursorpositionen: Durch gleichzeitiges Drücken der SHIFT-Taste und einer Funktionstaste wird die momentane Cursorposition gespeichert, durch Drücken einer Funktionstaste allein springt der Cursor wieder genau an die gespeicherte Stelle. F7 wird von den Befehlen L, R und C automatisch belegt.

H — Horizontale Schrittweite
V — Vertikale Schrittweite
F1 bis F8: Schrittweiten speichern

Noch eine Speicherfunktion haben die Funktionstasten: Normalerweise bewegt sich der Cursor in 1-Pixel-Schritten bei der Joysticksteuerung, beziehungsweise in 8-Pixel-Schritten bei Steuerung mittels Cursortasten. Diese Schrittweiten sind jedoch — getrennt für horizontale und vertikale Bewegung — frei programmierbar, vier Schrittweitenpaare können auf den Funktionstasten gespeichert werden. Das Anwählen einer Schrittweite erfolgt durch gleichzeitiges Drücken der C = (Commodore-)Taste und einer Funktionstaste (gilt nur für die Cursortastensteuerung, der Joystick holt seine Schrittweite immer aus F1), das Programmieren durch die Befehle H und V, wonach die aktuelle (angewählte) Schrittweite angezeigt wird und geändert werden kann.

Im Einschaltzustand sind die Tasten folgendermaßen belegt:
F1: H = 1, V = 1, das ist die normale Joysticksteuerung. Durch Vergrößerung auf 2 oder 3 lassen sich im Draw-Modus punktierte Linien zeichnen.
F3 (Diese Taste ist im Einschaltzustand angewählt): H = 8, V = 8, für Textmodus. Vergrößerung ergibt eine gedehnte Schrift oder einen größeren Zeilenabstand.
F5: H = 24, V = 21, = Spritemaße, zur flächendeckenden Bearbeitung mittels Sprite-Editor.
F7: H = 160, V = 96, zur Einteilung des Bildschirmes in Viertelbilder für Walk-Befehl.

Weitere Anwendungsmöglichkeiten sind: Maßstäbe, Gitterraster, exakt symmetrische Zeichnungen etc...

Space: Sprite-Editor

Durch Drücken der Leertaste kommt man in den Sprite-Editor. Das Setzen und Löschen von Punkten geht dort genauso wie im Draw-Modus. Die Befehle des Grafik-Editors sind im Sprite-Editor nicht zugänglich, dafür stehen die folgenden zur Verfügung:

M Mirror — Das Sprite wird zur Senkrechten gespiegelt.
T Turn — Drehung um 180 Grad. M und T ergeben eine Spiegelung zur Waagrechten.
R Rotate — Das Sprite wird um 90 Grad im Uhrzeigersinn gedreht. Da es jedoch 24 Punkte breit, aber nur 21 hoch ist, gehen die rechten drei Spalten verloren. Außerdem ist zweimal R nicht dasselbe wie T.
G Grid — Zum besseren Abzählen von Punkten wird ein Gitter eingeblendet, bei nochmaliger Eingabe von G wird es wieder ausgeblendet.

SHIFT CLR **Sprite löschen**
Space **Sprite-Editor verlassen**
W Walk **Bildfolge ablaufen lassen**
SHIFT W **Bildsequenz programmieren**

Jetzt lernen die Bilder laufen! Mit W werden die Bildschirmspeicher in schneller, programmierbarer Folge zyklisch durchgeschaltet. Da jedoch sechs Bilder für einen Bewegungsablauf recht wenig sind (Speicher 7 wird als »Leinwand« benutzt, sein Inhalt geht verloren) und außerdem die Bildfolge-Geschwindigkeit bei der Verschiebung von jedesmal 8 KByte nicht gerade hoch ist (maximal 7 Bilder/s), können die sechs Bildschirme in 24 Viertelbilder (mit je 160 x 96 Punkten, die unterste Zeile bleibt frei) zerlegt werden, die bei Maximalgeschwindigkeit in einer Sekunde »durchgerasselt« werden.

Bei der Erstellung solcher Viertelbilder ist die programmierbare Schrittweite (160/96) sowie der Move-Befehl besonders nützlich. Die Programmierung der Bildfolge geschieht durch einen Sequenz-String, der aus den Zahlen 1 bis 6 zum Aufruf der großen Bilder oder aus den Buchstaben A bis X zum Aufruf der Viertelbilder bestehen kann. Der Bildschirmspeicher 1 enthält die Viertelbilder A (links oben), B (rechts oben), C (links unten) und D (rechts unten). Das geht so weiter bis zum Speicher 6, der die Viertelbilder U, V, W, X enthält. Beispiel: Zum sequentiellen Durchschalten aller Viertelbilder besteht der Sequenzstring aus den Buchstaben A bis X in alphabetischer Reihenfolge. Zahlen und Buchstaben dürfen mehrfach und sogar gemischt vorkommen.

Mit SHIFT W gelangt man in den Sequenzstring-Editor, mit W wird die Bewegung gestartet. Während des Laufes kann die Geschwindigkeit mit der Plus- und Minus-Taste geregelt werden, bei gedrückter SHIFT-Taste läuft der Film rückwärts. Durch Druck auf den Feuerknopf wird die Vorführung beendet.

Bei Farb-Betrieb wird Bildschirm 6 als Leinwand verwendet, es stehen nur noch 5 große oder 20 kleine Bilder zur Verfügung.

Z Zeichensatz

HI-EDDI kann auch als Zeichensatz-Editor verwendet werden: Mittels Z wird der gerade angewählte Zeichensatz in die ersten sieben Zeilen des Grafikbildschirmes kopiert. Dort kann er mittels Get, Sprite-Editor und Stamp modifiziert, anschließend auf Diskette gespeichert und von anderen Programmen, zum Beispiel Textverarbeitungsprogramme, verwendet werden.

C= L LOAD (C = bedeutet Commodore-Taste)
C= S SAVE
C= D Directory anzeigen
C= C Commando an Disk oder (wenn nur RETURN) Fehlerkanal lesen

Bei LOAD und SAVE hat man die Wahl zwischen (schwarzweißem) Grafikbild, Farbbild, Zeichensatz und Sprite. LOAD und SAVE beziehen sich immer auf den aktuellen (= sichtbaren) Bildschirmspeicher (beziehungsweise Sprite). Bei SAVE Zeichensatz muß dieser in den ersten sieben Zeilen des sichtbaren Bildschirmes stehen. Bei LOAD Sprite sollte man wirklich nur ein Sprite und nichts längeres laden, sonst gibt's einen netten Absturz. Es können auch Diashow-Bilder und Files anderer Grafikprogramme gelesen werden.

C= P Print — Ausdrucken

Die Druckerroutine HI-PRINT wird bei Bedarf in Overlaytechnik nachgeladen, nach Beendigung des Druckvorganges wird wieder HI-EXE geladen.

Die vorliegende Druckerroutine bietet folgende Möglichkeiten:

1. Ausdruck eines Bildes, groß oder klein.
2. Zwei Bilder nahtlos nebeneinander, was natürlich nur in klein geht.

Superhardcopy

Vor und nach dem Ausdruck werden keine zusätzlichen Zeilenvorschübe ausgegeben, aufeinanderfolgende Ausdrucke hängen somit nahtlos aneinander (Bild 1). So läßt sich zum Beispiel mit sechs Bildern eine Superhardcopy mit 640 x 600 Punkten erstellen (Bild 2), lädt man zwischendurch weitere Bilder von Diskette nach, kann man meterlange Bilder ausdrucken.

Leider ist Drucker nicht gleich Drucker! Die vorliegende Druckerroutine (Listing 1) läßt sich nur für Epson-Drucker und kompatible mit Interface verwenden. Deshalb möchte ich im folgenden Kapitel Hinweise zur Erstellung einer Druckerroutine für andere Konfigurationen geben.

Die Speicherbelegung von HI-EDDI:

Adresse	Belegung
\$57-\$60 und \$F9-\$FE:	Temporäre Speicherzellen
\$2C0-\$2FE:	Residenter Datenbereich (Sprite-Inhalt)
\$340-\$3FE:	Temporärer Datenbereich (Kreuz-, großes und kleines Rahmen-Sprite
\$801-\$CFF:	Basic-Speicher
\$D00-\$1F4C:	Maschinenprogramm
\$1F80-\$1FFF:	Residenter Datenbereich (Betriebszustände etc.)
\$2000, \$4000, \$6000 etc.:	Grafikspeicher

Die Druckerroutine wird in denselben Bereich wie HI-EXE geladen, also ab $D00, nach Beendigung des Druckvorganges wird wieder HI-EXE geladen (Listing 1 und 2).

Die vorliegende Druckerroutine enthält auch ein Software-Interface, das den User-Port zur Centronics-Schnittstelle umfunktioniert. Dazu muß ein Kabel mit folgender Pinbelegung verwendet werden:

Pin am User-Port		Pin am Centronics-Stecker	
M	(PA2)	1	(Strobe)
C	(PB0)	2	(DATA 1)
D	(PB1)	3	(DATA 2)
E	(PB2)	4	(DATA 3)
F	(PB3)	5	(DATA 4)
H	(PB4)	6	(DATA 5)
J	(PB5)	7	(DATA 6)
K	(PB6)	8	(DATA 7)
L	(PB7)	9	(DATA 8)
B	(FLAG2)	10 oder 11	(ACKNLG oder BUSY)
A	(GND)	16	

Will man eine Routine für andere Drucker schreiben, so darf diese den gesamten Speicherbereich von $D00 bis $1F7F belegen, lediglich die residenten Speicherbereiche und der Basic-Speicher sind tabu. Die Information darüber, wo welcher Bildschirm im Speicher liegt, steht in den 7 Bytes von $1F96 bis $1F9C: In $1F96 steht die Nummer des Bildes, das ab $2000 in Speicher liegt (das ist das sichtbare Bild!), in Zelle $1F97 steht die Nummer des Bildes ab $4000 und so weiter. Beispiel: Soll Bild Nummer 2 ausgedruckt werden, so sucht man in der Tabelle nach der 2. Steht sie zum Beispiel in Zelle $1F9A, so liegt Bild 2 im Bereich ab $A000 (unter dem Basic-ROM). Im Farb-Betrieb steht in Zelle $1F97 eine 0, weil im Bereich $4000 bis $6000 dann die Farbinformationen für die sechs Bilder liegen und dieser Bereich für Grafikbilder somit gesperrt sein muß.

Für den nicht Maschinensprache-erfahrenen Leser sind dies sicherlich alles spanische Dörfer, aber vielleicht sind einige »Maschinen-Profis« so nett und schicken Lösungen für andere Drucker ein.

HI-EDDI als Menüprogramm

Wem die Eingabe der Befehle über die Tastatur nicht gefällt, der kann HI-EDDI zu einem Menüprogramm nach dem Vorbild des Koala Painter umfunktionieren, bei dem die Befehle über eine bunte, illustrierte Menütafel eingegeben werden. Ehrlich gesagt: Ich finde die Eingabe über Menütafel alles andere als benutzerfreundlich: Menü holen, Cursor auf das gewünschte Feld bugsieren und dann Knopf drücken ist eine wesentlich umständlichere Prozedur als ein einfacher Tastendruck. Dafür macht das Entwerfen und Realisieren einer Menütafel so viel Spaß, daß ich mich entschlossen habe, diese Möglichkeit vorzusehen.

Doch zunächst zurück zu dem Bytewert, der bei Programmstart auf die Frage »Betriebsart« eingegeben werden muß. Er setzt sich wie folgt zusammen:

Bit 7:	=0 :	Schwarzweiß-Betrieb
	=1 :	Farb-Betrieb
Bit 6 :	=0 :	nur Tastatureingabe
	=1 :	Menü-Eingabe
Bit 5-1 :		keine Funktion
Bit 0 :	=0 :	Beim Start werden alle Bildschirmspeicher gelöscht
	=1 :	Es wird nichts gelöscht, im Speicher befindliche Bilder bleiben erhalten

Für Farbe und Menü zum Beispiel muß 192 eingegeben werden. Ich finde die Menü-Eingabe nur in Farbe sinnvoll, ein Schwarzweiß-Menü sieht recht langweilig aus. Ist das Bit 6 gesetzt, also die Menü-Betriebsart angewählt, so muß sich auf der Diskette unter dem Namen »MENUE« die Menütafel befinden, die beim Start automatisch nachgeladen wird. Das Erstellen einer solchen Menütafel soll nun erklärt werden:

Die Menütafel wird in 20 x 12, also insgesamt 240 Felder von je 16 x 16 Punkten eingeteilt. 20 x 16 = 320, die Breite des Bildschirms, aber 12 x 16 = 192 gibt nicht ganz die Höhe des Bildschirmes: Die letzte Zeile (acht Punkte) bleibt, wie bei Walk, frei. In dieser Zeile werden die Steuerbytes untergebracht, für jedes der 240 Felder 2 Bytes. Wird im Menü-Modus der Cursor auf eines der Felder gefahren und der Feuerknopf gedrückt, dann holt sich HI-EDDI aus der letzten Zeile die diesem Feld zugeordneten Steuerbytes und entnimmt aus ihnen, welcher Befehl ausgeführt werden soll. Die Zuordnung von Feldern zu Steuerbytes geht zeilenweise, der Reihe nach: Die ersten beiden Steuerbytes (Adresse $3E00 und $3E01, wenn die Menütafel angewählt ist) gehören zum ersten Feld links oben, die nächsten beiden Bytes zu dem daneben und so weiter. Die Steuerbytes zum letzten Feld haben die Adressen $3FDE und $3FDF, das liegt bereits im unsichtbaren Teil des Bildschirmspeichers. Die nicht gerade ansehnlichen Steuerbytes im sichtbaren Teil können durch gleiche Farben für Vorder- und Hintergrund versteckt werden.

Somit ist klar, wie man vorgehen muß: Zunächst ist ein Menü-Bild zu zeichnen, bei dessen Einteilung nur die 20 x 12-Feld-Rasterung zu berücksichtigen ist. Wie man die Befehle anordnet, wieviele Felder man für einen Befehl zusammenhängt und welche Befehle man überhaupt ins Menü aufnimmt, bleibt jedem selbst überlassen. Denn keineswegs alle Befehle sind »menüfähig«: Die Korrekturtaste (Pfeil nach rechts) wirkt nur bis zum letzten Bildschirmwechsel zurück. Da jedoch das Anwählen der Menütafel ein Bildschirmwechsel ist, verliert die Korrekturtaste dabei ihre Wirkung. Dasselbe gilt für die Pfeil-nach-oben-Taste: Die Anwahl der Menütafel gilt als Moduswechsel! Aus demselben Grund muß beim MOVEn über zwei Seiten die Seitennummer über die Tastatur eingegeben werden, da bei einem Moduswechsel die Markierung eines Quellbereiches gelöscht wird.

Aus dem Textmodus kommt man nicht direkt ins Menü (SPACE wird als zu druckendes Zeichen interpretiert). Es muß daher zuerst ein anderer Modus über die Tastatur angewählt werden (zum Beispiel CTRL D für Draw), bevor das Menü wieder zugänglich ist.

Dafür bietet die Menütafel auch einige Vorteile: Die Farbenwahl erfolgt nicht mehr durch umständliches Fortschalten, sondern durch Direktanwahl der Farben. Und die Befehle U, O, X, die normalerweise zwei Eingaben erfordern, werden jetzt mit einer Eingabe »erschlagen«.

Hat man nun ein Menübild gemalt (und abgespeichert!), so folgt der zweite Teil, das Erstellen der Steuerbytes, am besten in Form eines DATA-Laders (siehe Listing 4). Von den zwei Steuerbytes pro Feld enthält das zweite den Tastaturcode des jeweiligen Befehls. Die Tastaturcodes sind im 64'er, Ausgabe 5/84, Seite 105 zusammengestellt, sie lassen sich auch mit dem folgenden »Programm« ermitteln:

```
10 PRINT PEEK(203):GOTO10
```

Beim ersten Steuerbyte muß man zwischen verschiedenen Befehlskategorien unterscheiden:

1. »Normale« Befehle: Bit 0 : SHIFT-Bit, muß gesetzt sein, wenn für den entsprechenden Befehl die SHIFT-Taste gedrückt werden muß. Bit 1 : C=-Bit, muß bei gedrückter Commodore-Taste gesetzt sein. Die Bits 2 bis 7 müssen 0 sein. Beispiel: Sprite-Editor einschalten ergibt die Bytes 1 (SHIFT, im Menü-Betrieb dient Space zur Anwahl der Menütafel, der Sprite-Editor ist nur mehr über SHIFT-Space erreichbar) und 60 (Tastaturcode Space).

2. Befehle U, O, X: Tastaturcode der entsprechenden Nummerntaste. Beispiel: X 5 ergibt die Steuerbytes 16 (Tastaturcode »5« und 23 (Tastaturcode »X«).
3. Farbbefehle: Bit 0 : SHIFT-Bit. Bit 2 bis 5: Nummer der Farbe, wie im Commodore-Handbuch angegeben. Die Bits 1 und 6 müssen 0 sein, das Bit 7 ist 1. Beispiel: Total Foreground, Blau ergibt: 128 (Bit 7) + 4 x 6 (blau) + 1 (SHIFT) = 153 und 21 (»F«).
4. »Leerer« Befehl ohne Verlassen der Menütafel (zum Beispiel für Überschrift): 1. Byte = 64, 2. Byte = 0
5. »Leerer« Befehl mit Verlassen des Menüs: 1. Byte = 0, 2. Byte = 64.

Die 5 Byte im Anschluß an die 480 Steuerbytes müssen die Kennung »CBM80« enthalten. Das ist eigentlich die ROM-Kennung, ich habe sie hier als Menütafel-Kennung mißbraucht, um versehentliche Menü-Zugriffe auf ein normales Bild zu vermeiden. Vorsicht: Fehler in den Steuerbytes können zum Programmabsturz führen!

Nun braucht man nur noch Menübild und Steuerbytes zu verknubbeln: Dazu HI-EDDI (Listing 3) laden, Starten (Betriebsart 128), Menübild laden und Programm mit STOP/RESTORE unterbrechen. Steuerbyte-DATA-Lader laden und starten. Der Lader muß, wie im Beispiel (Listing 4) zu sehen,mit den Befehlen POKE56, 32:CLR beginnen, um das im Speicher befindliche Menübild zu schützen! Dann HI-EDDI laden, starten (Betriebsart 129, damit die jetzt bereits fertige Menütafel im Speicher nicht gelöscht wird) und die nun sichtbare Menütafel unter dem Namen »MENUE« speichern.

Wie wird der »Menü-HI-EDDI« nun bedient? Beim Start muß auf die Frage Betriebsart 192 (oder 193) eingegeben werden. Auf der Menütafel fährt man den Cursor auf den gewünschten Befehl und drückt den Feuerknopf. Daraufhin wird das Menü verlassen, auf den aktuellen Bildschirm zurückgeschaltet (im Einschaltzustand ist das jetzt 2, da in Speicher 1 die Menütafel steckt) und der Befehl ausgeführt. Das Menü wird durch Drücken der Space-Taste auf den Bildschirm geholt. Die Tastatureingabe bleibt neben dem Menü voll erhalten. Wie schon erwähnt, belegt das Menü den Speicher 1. Dieser Speicher kann natürlich auch direkt angewählt werden, allerdings befindet man sich dann nicht im Menü-Modus, sondern die Menütafel wird wie ein ganz normales Bild behandelt!

Zum Abschluß noch ein Tip: HI-EDDI läßt sich mit STOP/RESTORE unterbrechen (nicht während einer Befehlsausführung!), ein Warmstart ist mit GOTO150 möglich, ein Kaltstart ohne Nachladen von HI-EXE, aber mit Betriebsart-Eingabe mit RUN100.

(Hans Haberl/ah)

```
10 REM*********************************       <060>
20 REM*                               *       <069>
30 REM*            HI-EDDI            *       <137>
40 REM*        VON HANS HABERL        *       <158>
50 REM*                               *       <099>
60 REM*   DATA-LADER ZUR ERZEUGUNG    *       <170>
70 REM*DER DRUCKERROUTINE "HI-PRINT",*       <043>
80 REM*  WIRD AUF DISKETTE ABGELEGT   *       <222>
90 REM*********************************       <140>
100 :                                         <076>
110 REM  CHECKSUMMEN, NICHT VERTIPPEN !!      <083>
120 DATA-205,164,-457,322,689,-834,-49,290
    ,-87,17                                   <239>
130 :                                         <106>
140 REM M-CODE PROGRAMM                       <012>
150 DATA 0,13,133,91,173,105,14,240,34,120
    ,169,255,141,3,221,173,2,221,9,4          <206>
160 DATA 141,2,221,173,0,221,9,4,141,0,221
    ,169,16,141,13,221,173,13,221,88,-1       <145>
170 DATA 76,68,13,32,204,255,169,4,174,106
    ,14,172,107,14,32,186,255,169,0           <230>
180 DATA 32,189,255,32,192,255,162,4,32,20
    1,255,169,4,133,94,160,3,32,34,14,-1      <117>
190 DATA 169,128,133,95,169,2,133,96,165,9
    1,41,192,208,4,70,95,70,96,165,91         <242>
200 DATA 41,7,162,0,32,48,14,165,91,48,3,7
    4,74,74,41,7,162,2,32,48,14,169,-1        <055>
210 DATA 25,133,92,165,95,48,10,160,20,169
    ,32,32,71,14,136,208,250,160,8,32         <210>
220 DATA 34,14,165,95,32,71,14,165,96,32,7
    1,14,169,40,133,93,120,169,52,133,-1      <024>
230 DATA 1,160,7,177,87,153,125,14,136,16,
    248,169,55,133,1,88,165,87,24,105         <139>
240 DATA 8,133,87,144,2,230,88,160,8,162,0
    ,62,125,14,8,42,232,40,36,91,48,-1        <112>
250 DATA 6,224,8,208,241,240,8,42,228,94,2
    08,234,32,71,14,32,71,14,136,208          <176>
260 DATA 223,198,93,208,186,165,95,16,38,1
    62,1,180,87,181,89,149,87,148,89,-1       <143>
270 DATA 202,16,245,165,94,73,12,133,94,20
    1,4,240,15,165,91,48,3,76,151,13          <171>
280 DATA 160,0,32,34,14,76,122,13,198,92,2
    08,244,160,0,32,34,14,160,13,32,-1        <184>
290 DATA 34,14,32,204,255,169,4,32,195,255
    ,96,185,108,14,201,255,240,6,32           <067>
300 DATA 71,14,200,208,243,96,160,6,217,15
    0,31,240,3,136,208,248,169,0,149,-1       <073>
310 DATA 87,24,105,32,136,16,251,149,88,96
    ,174,105,14,208,3,76,210,255,72           <035>
320 DATA 141,1,221,173,0,221,41,251,141,0,
    221,9,4,141,0,221,173,13,221,41,-1        <158>
330 DATA 16,240,249,104,96,-1                 <178>
340 :                                         <062>
350 REM DIE FOLGENDEN DATAS MUESSEN AN        <058>
360 REM DEN DRUCKER ANGEPASST WERDEN!!        <039>
370 REM (DIE ANGEGEBENEN WERTE SIND           <151>
380 REM FUER EINEN EPSON RX-80 MIT            <246>
390 REM DATA BECKER INTERFACE)                <086>
400 :                                         <122>
410 REM 0=SER. BUS, 1=USERPORT                <143>
420 DATA 0                                    <053>
430 REM GERAETEADRESSE                        <076>
440 DATA 4                                    <105>
450 REM SEKUNDAERADRESSE (DIREKTMODUS)        <254>
460 DATA 1                                    <101>
470 REM DIE LAENGE DER FOLGENDEN DATA-        <247>
480 REM ZEILEN DARF NICHT VERAENDERT          <162>
490 REM WERDEN, GGF. MIT 255 AUFFUELLEN       <136>
500 REM (MIND. EIN 255 MUSS IN JEDER          <077>
510 REM ZEILE BLEIBEN, IST ENDEKENNZ.)        <175>
520 REM CARRIAGE RETURN LINE FEED             <156>
530 DATA 13,10,255                            <214>
540 REM ZEILENABSTAND FUER GRAFIK             <158>
550 DATA 27,51,23,255,255                     <255>
560 REM CRT-GRAFIK (640 PUNKTE/ZEILE)         <097>
570 REM (OHNE BYTE-ANZAHLEN!)                 <150>
580 DATA 27,42,4,255,255                      <150>
590 REM NORMALER ZEILENABSTAND                <235>
600 DATA 27,50,255,255                        <084>
610 DATA-2:REM DATA-ENDE                      <115>
620 :                                         <088>
630 REM CHECKSUMMENPRUEFUNG                   <215>
640 V=1:FOR I=0 TO 9:READ S(I):NEXT           <079>
650 FOR B=0 TO 9                              <156>
660 READ A:IF A>=0 THEN S=S+A*V:V=-V:GOTO
    660                                       <183>
670 IF S<>S(B)THEN PRINT"DATA-FEHLER IN ZE
    ILE"150+20*B"ODER"160+20*B:END            <223>
680 S=0:NEXT:PRINT"DATAS OK"                  <004>
690 :                                         <158>
700 REM FILE-ABLAGE                           <080>
710 RESTORE:FOR I=0 TO 9:READ A:NEXT          <007>
720 OPEN 2,8,2,"HI-PRINT,P,W"                 <027>
730 READ A:IF A>=0 THEN PRINT#2,CHR$(A);      <226>
740 IF A>-2 THEN 730                          <108>
750 CLOSE 2:END                               <184>

© 64'er
```

Listing 1. HI-PRINT ist eine ausgezeichnete Druckerroutine für den FX-80 und kompatible Drucker. Bitte beachten Sie die Eingabehinweise auf Seite 16.

```
programm : hi-exe                   0d00 1f4c
------------------------------------------

0d00 : 4c 11 1b 4c c1 1e 4c 36  d0
0d08 : 1f 20 74 1c 20 24 0d ad  8b
0d10 : 83 1f 20 d3 1c 20 67 12  2a
0d18 : 20 d1 13 ee 28 d0 ee 29  db
0d20 : d0 4c 0c 0d a5 cb cd 80  ac
0d28 : 1f f0 28 8d 80 1f c9 40  24
0d30 : f0 21 a8 ad 8d 02 8d 81  b3
0d38 : 1f aa 29 04 d0 0a ad 84  94
0d40 : 1f c9 01 d0 03 4c 12 10  99
0d48 : 98 a2 2a dd 6e 0d f0 04  93
0d50 : ca 10 f8 60 8a ae 84 1f  db
0d58 : e0 0c d0 05 48 20 0d 11  ef
0d60 : 68 0a aa bd 9a 0d 48 bd  de
0d68 : 99 0d 48 8a 4a 60 33 38  d0
0d70 : 3b 08 0b 10 13 18 1d 1f  19
0d78 : 1c 15 04 05 06 03 12 16  ae
0d80 : 2a 11 14 29 24 09 0e 1a  54
0d88 : 0d 0a 21 26 17 1e 0c 20  7a
0d90 : 23 02 07 00 3c 39 36 28  2d
0d98 : 2b 54 0e 63 0e 63 0e 63  d8
0da0 : 0e 63 0e 63 0e 63 0e 63  4b
0da8 : 0e f7 0d f7 0d ed 0e ed  48
0db0 : 0e ba 0f ba 0f ba 0f ba  af
0db8 : 0f ee 0d 06 0e ee 0d 06  db
0dc0 : 0e ee 0d ee 0d 06 0e 23  e6
0dc8 : 11 06 0e 06 0e ee 0d 06  b9
0dd0 : 0e 3d 0e bf 0e c5 0e b9  b3
0dd8 : 0e b2 10 00 10 04 10 bc  1e
0de0 : 12 3a 13 fa 12 e2 10 73  93
0de8 : 1c f4 18 f6 0f fa 0f aa  be
0df0 : ad 81 1f 29 02 f0 0f 8a  44
0df8 : 29 0f 8d 80 1f 20 47 1c  65
0e00 : 20 e1 1c 68 68 60 8a 29  2b
0e08 : 0f 8d 84 1f aa bd 24 0e  28
0e10 : 8d f9 07 bd 31 0e 8d 15  f7
0e18 : d0 a9 00 8d aa 1f 8d a9  9c
0e20 : 1f 4c 5f 13 0d 0e 0d 0d  2f
0e28 : 0d 0d 0d 0d 0f 0f 0f 0f  65
0e30 : 0d 02 02 02 02 02 02 02  3b
0e38 : 02 02 02 03 03 02 a2 20  27
0e40 : 86 58 a0 00 84 57 b1 57  93
0e48 : 49 ff 91 57 c8 d0 f7 e6  a1
0e50 : 58 ca d0 f2 60 ad 81 1f  57
0e58 : 29 01 f0 f8 a9 51 8d a9  0c
0e60 : 1f ad 96 1f 20 cc 0e 48  11
0e68 : 86 fc 78 a9 34 85 01 ad  8e
0e70 : a9 1f d0 29 b1 57 aa b1  e6
0e78 : 59 91 57 8a 91 59 c8 d0  6a
0e80 : f3 e6 58 e6 5a c6 fc d0  4b
0e88 : eb 68 20 80 0f a8 ae 96  de
0e90 : 1f b9 96 1f 8d 96 1f 8a  34
0e98 : 99 96 1f d0 17 8d a2 0e  e3
0ea0 : b1 59 31 57 91 57 c8 d0  ce
0ea8 : f7 e6 58 e6 5a ca d0 f0  26
0eb0 : 8e a9 1f 68 a9 37 85 01  54
0eb8 : 58 60 a9 31 8d a9 1f 60  34
0ec0 : a9 11 8d a9 1f 60 a9 51  c9
0ec8 : 8d a9 1f 60 a0 06 d9 96  cd
0ed0 : 1f f0 03 88 d0 f8 98 48  01
0ed8 : 18 a9 00 69 20 88 10 fb  71
0ee0 : 85 5a a2 20 86 58 a0 00  ed
0ee8 : 84 57 84 59 68 60 29 01  94
0ef0 : 85 fd ad 89 1f 29 fe 05  52
0ef8 : fd 8d 89 1f aa ad 81 1f  5e
0f00 : 29 01 d0 0f ee 20 d0 ee  d1
0f08 : 27 d0 8a 10 05 a9 07 20  36
0f10 : 07 0e 60 ad f9 07 48 a2  2a
0f18 : 04 86 58 a0 00 84 57 18  3b
0f20 : 20 5f 1a c8 d0 f9 e6 58  b9
0f28 : ca d0 f4 f0 3e ad f9 07  fd
0f30 : 48 ad 88 1f a2 04 86 58  6a
0f38 : a0 00 84 57 91 57 c8 d0  7d
0f40 : fb e6 58 ca d0 f6 f0 23  ec
0f48 : ae 89 1f 10 e0 a9 40 85  ec
0f50 : 58 a2 04 86 5a d0 0d ae  89
0f58 : 89 1f 10 1d a9 40 85 5a  80
0f60 : a2 04 86 58 ad f9 07 48  08
0f68 : 20 c9 10 68 8d f9 07 a9  96
0f70 : 0b 8d f8 07 a9 0d 8d fa  90
0f78 : 07 ae 00 04 8e 88 1f 60  c1
0f80 : 48 ad 84 1f c9 06 d0 0a  c8
0f88 : ad aa 1f c9 02 d0 03 20  7e
0f90 : 48 0f ad 89 1f 10 18 ad  2b
0f98 : 96 1f 20 b1 0f 20 5e 0f  86
0fa0 : 68 48 a8 b9 96 1f 20 b1  d4
0fa8 : 0f 20 4f 0f 20 57 0f 68  47
0fb0 : 60 a8 a9 40 18 69 04 88  c5
0fb8 : d0 fb 60 29 03 aa ad 81  02
0fc0 : 1f 29 03 c9 01 f0 19 29  bc
0fc8 : 02 d0 28 bd 9d 1f 8d 85  08
0fd0 : 1f bd a1 1f 8d 86 1f bd  1f
0fd8 : a5 1f 8d 87 1f 4c 5f 13  59
0fe0 : ad 85 1f 9d 9d 1f ad 86  62
0fe8 : 1f 9d a1 1f ad 87 1f 9d  f1
0ff0 : a5 1f 60 8e 8a 1f 60 a9  85
0ff8 : 06 d0 02 a9 0c 8d 94 1f  da
1000 : 60 a2 80 d0 02 a2 00 ad  7c
1008 : 81 1f 29 04 f0 03 8e 95  70
1010 : 1f 60 e0 03 f0 fb 8a 0a  25
1018 : aa bd 79 eb 85 57 bd 7a  7c
1020 : eb 85 58 b1 57 aa 29 7f  89
1028 : c9 20 b0 06 ac 80 1f 4c  d2
1030 : 48 0d 8a 29 e0 c9 60 d0  46
1038 : 04 a9 40 d0 09 8a 10 04  68
1040 : 09 40 d0 02 29 bf 29 7f  12
1048 : 85 fd 8a 29 1f 05 fd 0d  c0
1050 : 95 1f a2 00 86 58 0a 26  bd
1058 : 58 0a 26 58 0a 26 58 85  88
1060 : 57 ad 18 d0 29 02 0a 0a  8d
1068 : 09 d0 05 58 85 58 78 a9  76
1070 : 33 85 01 a0 07 b1 57 99  49
1078 : ba 1f 88 10 f8 a9 37 85  ab
1080 : 01 58 20 11 15 a2 00 20  7e
1088 : be 14 a9 08 85 5f a0 00  92
1090 : 3e ba 1f a9 00 90 02 a5  00
1098 : 5e 11 57 91 57 20 06 15  40
10a0 : c6 5f d0 ec e6 5b e8 e0  96
10a8 : 08 d0 dc a9 00 8d 81 1f  35
10b0 : 4c bd 12 ad 18 d0 29 02  c6
10b8 : 0a 0a 09 d0 85 58 a9 20  26
10c0 : 85 5a a2 08 78 a9 33 85  c9
10c8 : 01 a0 00 84 57 84 59 b1  0c
10d0 : 57 91 59 c8 d0 f9 e6 58  88
10d8 : e6 5a ca d0 f2 a9 37 85  1c
10e0 : 01 58 60 ad 89 1f 49 40  12
10e8 : 29 40 0d 81 1f f0 03 4c  c3
10f0 : 27 1d a0 03 b9 84 1f 99  9e
10f8 : af 1f 88 10 f7 ad 96 1f  e0
1100 : 8d ae 1f a9 01 20 64 0e  a0
1108 : a9 0c 4c 07 0e ad ae 1f  f3
1110 : 20 64 0e a0 03 b9 af 1f  f5
1118 : 99 84 1f 88 10 f7 ad 84  4d
1120 : 1f 4c 07 0e ad 81 1f 29  9f
1128 : 01 f0 03 4c f8 0d ad 15  c5
1130 : d0 48 a2 00 8e 15 d0 8e  bf
1138 : a9 1f 86 fc e8 86 fd ad  c8
1140 : 89 1f 2a 2a 29 01 49 07  f7
1148 : 20 64 0e a0 04 b1 2d 99  e8
1150 : 59 00 88 d0 f8 c6 5b 10  39
1158 : 01 60 a4 fc b1 5c ae 8d  26
1160 : 02 f0 0a c6 fc 10 10 a6  14
1168 : 5b 86 fc d0 0a e6 fc c4  b5
1170 : 5b 90 04 a2 00 86 fc c9  24
1178 : 40 b0 2c 29 07 8d ae 1f  16
1180 : 20 cc 0e 78 a9 34 85 01  ed
1188 : b1 59 91 57 c8 d0 f7 e6  fe
1190 : 58 e6 5a ca d0 f2 ad 89  ba
1198 : 1f 10 09 ad ae 1f 20 b1  7f
11a0 : 0f 20 4f 0f 4c 3d 12 38  dd
11a8 : e9 01 48 4a 4a 29 07 18  a7
11b0 : 69 01 8d ae 1f 20 cc 0e  15
11b8 : 68 48 4a 90 04 a2 a0 86  ce
11c0 : 59 4a 90 06 a5 5a 09 0f  93
11c8 : 85 5a a9 27 85 58 a9 d0  2d
11d0 : 85 57 78 a9 34 85 01 a2  0d
11d8 : 0c a0 a0 88 b1 59 91 57  48
11e0 : 98 d0 f8 a5 57 18 69 40  2f
11e8 : 85 57 a5 58 69 01 85 58  f3
11f0 : a5 59 18 69 40 85 59 a5  56
11f8 : 5a 69 01 85 5a ca d0 d9  eb
1200 : 68 aa ad 89 1f 10 36 ad  01
1208 : ae 1f 20 b1 0f 85 58 a9  56
1210 : 00 85 57 8a 4a 90 04 a2  78
1218 : 14 86 57 4a 90 0d a5 57  45
1220 : 18 69 e0 85 57 a5 58 69  ac
1228 : 01 85 58 a9 fa 85 59 a9  cc
1230 : 04 85 5a a2 0b a0 13 8c  fd
1238 : c5 1f 20 1f 1a a9 37 85  50
1240 : 01 58 a5 fd 20 d3 1c a5  f3
1248 : cb c9 2b d0 03 38 26 fd  63
1250 : c9 28 d0 02 46 fd ad 00  ad
1258 : dc 29 10 f0 03 4c 5a 11  09
1260 : 68 8d 15 d0 4c e8 1c 20  ab
1268 : a2 12 29 0f d0 06 a9 28  74
1270 : 8d 83 1f 60 48 29 03 f0  4e
1278 : 0a 29 01 8d 81 1f a2 00  a4
1280 : 20 3e 13 68 29 0c f0 0c  60
1288 : 4a 4a 29 01 8d 81 1f a2  08
1290 : 00 20 c0 12 ae 83 1f ec  70
1298 : 94 1f 90 05 ca ca 8e 83  c5
12a0 : 1f 60 78 ae 02 dc a0 00  6d
12a8 : 8c 02 dc ad 00 dc cd 00  40
12b0 : dc d0 f8 8e 02 dc 58 49  ff
12b8 : ff 8d 82 1f 60 ae 8a 1f  e6
12c0 : ac 86 1f ad 81 1f 29 01  e5
12c8 : d0 1a 18 98 7d 8b 1f 90  90
12d0 : 24 ee 87 1f ae 87 1f e0  96
12d8 : 02 90 1a a9 01 8d 87 1f  b7
12e0 : a9 f0 b0 11 38 98 fd 8b  a7
12e8 : 1f b0 0a ce 87 1f 10 05  77
12f0 : a9 00 8d 87 1f 8d 86 1f  a4
12f8 : 4c 5f 13 ad 84 1f c9 01  d9
1300 : d0 38 ad 81 1f 29 01 49  5a
1308 : 01 8d 81 1f f0 03 20 bd  37
1310 : 12 20 11 15 a2 00 20 be  41
1318 : 14 a9 08 85 5f a0 00 a5  fa
1320 : 5e 49 ff 31 57 91 57 20  e8
1328 : 06 15 c6 5f d0 f1 e6 5b  45
1330 : e8 e0 08 d0 e1 ad 81 1f  74
1338 : f0 83 60 ae 8a 1f ac 85  37
1340 : 1f ad 81 1f 29 01 d0 0b  6e
1348 : 18 98 7d 8f 1f 90 0d a9  fc
1350 : ff b0 09 38 98 fd 8f 1f  e7
1358 : b0 02 a9 00 8d 85 1f 20  35
1360 : bb 13 bd c5 13 cd 85 1f  c1
1368 : 90 03 8d 85 1f bd c6 13  af
1370 : cd 85 1f b0 03 8d 85 1f  cf
1378 : ad 87 1f d0 0d bd c7 13  cf
1380 : cd 86 1f 90 10 8d 86 1f  30
1388 : b0 0b bd c8 13 cd 86 1f  3e
1390 : b0 03 8d 86 1f ad 85 1f  aa
1398 : 8d 01 d0 8d 03 d0 ad 86  06
13a0 : 1f 8d 00 d0 8d 02 d0 ad  27
13a8 : 87 1f 0a 0d 87 1f 85 fd  67
13b0 : ad 10 d0 29 fc 05 fd 8d  ca
13b8 : 10 d0 60 ad f9 07 38 e9  8b
13c0 : 0d 0a 0a aa 60 28 ef 0e  cd
13c8 : 4d 2b eb 11 49 32 e5 18  b6
13d0 : 40 ad 82 1f 29 10 f0 10  62
13d8 : 20 11 15 ad 84 1f 0a aa  3b
13e0 : bd ea 13 48 bd e9 13 48  e8
13e8 : 60 02 14 e7 13 68 15 0d  2e
13f0 : 16 55 16 9b 17 74 18 42  a4
13f8 : 1a 18 14 86 14 42 14 45  22
1400 : 14 95 1a 20 be 14 ac 8d  c4
1408 : 02 f0 09 a0 00 49 ff 31  85
1410 : 57 91 57 60 11 57 91 57  d2
1418 : 60 a5 5b 48 a2 00 20 be  53
1420 : 14 a9 18 85 5f a0 00 a5  06
1428 : 5e 49 ff 31 57 91 57 20  f0
1430 : 06 15 c6 5f d0 f1 e6 5b  4d
1438 : e8 e8 e8 e0 3f d0 df 68  b5
1440 : 85 5b 60 20 19 14 a2 00  4c
1448 : 20 be 14 a9 18 85 5f a0  6e
1450 : 00 bd c2 02 8d bc 1f bd  d6
1458 : c1 02 8d bb 1f bd c0 02  dc
1460 : 8d ba 1f 2e bc 1f 2e bb  cd
1468 : 1f 2e ba 1f a9 00 90 02  12
1470 : a5 5e 11 57 91 57 20 06  d4
1478 : 15 c6 5f d0 e6 e6 5b e8  c7
1480 : e8 e8 e0 3f d0 c2 60 a9  f4
1488 : 03 8d 15 d0 a2 00 20 be  d9
1490 : 14 a9 18 85 5f a0 00 18  5b
1498 : b1 57 25 5e f0 01 38 3e  7e
14a0 : c2 02 3e c1 02 3e c0 02  44
14a8 : 20 06 15 c6 5f d0 e8 e6  d7
14b0 : 5b e8 e8 e8 e0 3f d0 d6  cf
14b8 : a9 0b 8d 84 1f 60 a9 01  78
14c0 : 85 58 a5 5b 29 f8 85 57  65
14c8 : 0a 26 58 0a 26 58 18 65  8d
14d0 : 57 90 02 e6 58 0a 26 58  ec
14d8 : 0a 26 58 0a 26 58 85 57  37
14e0 : a5 5b 29 07 85 60 a5 5c  09
14e8 : 29 f8 18 65 60 65 57 85  da
14f0 : 57 a5 5d 65 58 85 58 a5  7c
14f8 : 5c 29 07 a8 a9 00 38 6a  10
1500 : 88 10 fc 85 5e 60 46 5e  3f
1508 : 90 06 66 5e 98 69 08 a8  47
1510 : 60 20 bb 13 ad 85 1f 38  c6
1518 : fd c5 13 85 5b ad 86 1f  e9
1520 : 38 fd c7 13 85 5c ad 87  ac
1528 : 1f e9 00 85 5d 60 20 e8  18
1530 : 1c ad aa 1f 49 01 8d aa  da
1538 : 1f f0 2d a2 02 b5 5b 9d  e5
1540 : ab 1f ca 10 f8 a2 03 20  20
1548 : e0 0f ad 02 d0 8d 04 d0  87
1550 : ad 03 d0 8d 05 d0 ad 10  12
1558 : d0 6a 08 2a 20 2a 8d 10  cf
1560 : d0 a9 06 8d 15 d0 68 68  82
1568 : 60 20 2e 15 20 03 14 a5  bc
1570 : 5b 48 a5 5c 48 a5 5d 48  9c
1578 : ad ac 1f 38 e5 5c 48 ad  08
1580 : ad 1f e5 5d 8d bd 1f b0  86
1588 : 0b 68 49 ff 69 01 48 a9  2d
1590 : 00 ed bd 1f 8d bb 1f 8d  28
1598 : bf 1f 68 8d ba 1f 8d be  0b
15a0 : 1f ad ab 1f 18 e5 5b 90  a4
15a8 : 04 49 ff 69 fe 8d bc 1f  0b
15b0 : 6e bd 1f 38 ed ba 1f aa  52
15b8 : a9 ff ed bb 1f 85 fc b0  c7
15c0 : 06 0a 0a 2a 20 74 17 ad  f1
15c8 : be 1f 6d bc 1f 8d be 1f  a0
15d0 : ad bf 1f e9 00 4c f5 15  c6
15d8 : ad bd 1f b0 e4 0a 2a 0a  9d
15e0 : 49 02 20 5c 17 18 ad be  24
15e8 : 1f 6d ba 1f 8d be 1f ad  f7
15f0 : bf 1f 6d bb 1f 8d bf 1f  ad
15f8 : 08 20 03 14 28 e8 d0 d8  12
1600 : e6 fc d0 d4 68 85 5d 68  2c
```

Listing 2. HI-EXE Maschinenroutine für HI-EDDI. Bitte beachten Sie die Eingabehinweise auf Seite 18.

```
1608 : 85 5c 68 85 5b 60 20 2e  1c
1610 : 15 ad ab 1f 48 a5 5b 8d  05
1618 : ab 1f 20 6c 15 68 8d ab  0b
1620 : 1f a5 5c 48 a5 5d 48 ad  f4
1628 : ac 1f 85 5c ad ad 1f 85  20
1630 : 5d 20 6c 15 68 85 5d 68  54
1638 : 85 5c a5 5b 48 ad ab 1f  9f
1640 : 85 5b 20 6c 15 68 85 5b  6a
1648 : a5 5c 8d ac 1f a5 5d 8d  c4
1650 : ad 1f 20 6c 15 60 20 2e  54
1658 : 15 a2 02 b5 5b 9d c2 1f  e1
1660 : ca 10 f8 a9 00 20 ea 16  7e
1668 : 20 03 14 20 b8 16 85 f9  59
1670 : 20 5a 17 08 a9 02 20 ea  85
1678 : 16 28 2a 0a 45 f9 20 5c  cc
1680 : 17 20 72 17 08 a9 04 20  45
1688 : ea 16 28 2a 45 f9 20 74  5a
1690 : 17 38 ad bc 1f ed be 1f  61
1698 : ad bd 1f ed bf 1f b0 06  6d
16a0 : 20 5a 17 4c a9 16 20 72  ed
16a8 : 17 90 0c a2 02 b5 5b dd  56
16b0 : c2 1f d0 b4 ca 10 f6 60  96
16b8 : 38 a5 5c ed ac 1f a8 a5  49
16c0 : 5d ed ad 1f 08 26 fd 28  5d
16c8 : 98 b0 04 49 ff 69 01 8d  4d
16d0 : c1 1f 38 ad ab 1f e5 5b  e7
16d8 : 08 26 fd 28 b0 04 49 ff  c8
16e0 : 69 01 8d c0 1f a5 fd 49  ef
16e8 : ff 60 48 20 b8 16 ad c0  a2
16f0 : 1f a2 00 20 36 17 ad c1  bb
16f8 : 1f a2 02 20 36 17 68 aa  00
1700 : 18 a5 57 65 59 9d ba 1f  19
1708 : a5 58 65 5a 9d bb 1f 8a  c7
1710 : f0 23 38 bd ba 1f ed ba  29
1718 : 1f 48 bd bb 1f ed bb 1f  d1
1720 : b0 0c a8 68 49 ff 69 01  4a
1728 : 48 98 49 ff 69 00 9d bb  93
1730 : 1f 68 9d ba 1f 60 48 a8  a9
1738 : a9 08 85 fc a9 00 95 57  86
1740 : 16 57 36 58 98 0a a8 90  38
1748 : 0b 68 48 18 75 57 95 57  b4
1750 : 90 02 f6 58 c6 fc d0 e8  13
1758 : 68 60 a5 f9 29 02 f0 08  0f
1760 : a5 5b f0 36 c6 5b 38 60  9f
1768 : a5 5b c9 c7 f0 2c e6 5b  e9
1770 : 38 60 a5 f9 29 01 f0 10  ff
1778 : a5 5c 05 5d f0 1c a5 5c  78
1780 : d0 02 c6 5d c6 5c 38 60  a0
1788 : a5 5d f0 06 a5 5c c9 3f  bc
1790 : f0 08 e6 5c d0 02 e6 5d  3d
1798 : 38 60 18 60 20 e8 1c a9  20
17a0 : 01 20 74 17 a9 00 85 f9  56
17a8 : a5 5b 8d ab 1f ba 8e bd  51
17b0 : 1f ba e0 14 90 6c a5 fd  e6
17b8 : 48 a5 fe 48 a5 5c 48 a5  45
17c0 : 5d 48 a5 5b 85 fd ad ab  6c
17c8 : 1f 85 fe 20 72 17 90 46  1c
17d0 : 20 23 18 90 41 a5 5b 48  d9
17d8 : ad ab 1f 48 20 23 18 90  c8
17e0 : 05 20 b1 17 90 f6 68 85  b2
17e8 : fe 68 85 fd a5 f9 49 01  8d
17f0 : 85 f9 20 72 17 20 23 18  f7
17f8 : 90 05 20 b1 17 90 f6 a5  66
1800 : f9 49 01 85 f9 20 72 17  27
1808 : ad 00 dc 29 10 d0 bc ae  e9
1810 : bd 1f 9a 4c e8 1c 68 85  a9
1818 : 5d 68 85 5c 68 85 fe 68  16
1820 : 85 fd 60 a5 fd 85 5b a5  35
1828 : fe 8d ab 1f 20 be 14 a0  45
1830 : 00 31 57 d0 36 a5 5b f0  99
1838 : 0d c6 5b 20 be 14 a0 00  92
1840 : 31 57 f0 f1 e6 5b a5 5b  2e
1848 : aa 20 be 14 a0 00 31 57  b2
1850 : d0 0e a5 5e 11 57 91 57  1d
1858 : e6 5b a5 5b c9 c8 d0 e9  bb
1860 : c6 5b a5 5b 8d ab 1f 86  68
1868 : 5b 38 60 e6 5b ad ab 1f  e4
1870 : c5 5b b0 b8 60 20 e8 1c  09
1878 : ad aa 1f c9 03 d0 05 a9  99
1880 : 00 8d aa 1f 0a aa a5 5c  1b
1888 : 46 5d 6a 4a 4a 9d bb 1f  1f
1890 : a5 5b 4a 4a 4a 9d ba 1f  79
1898 : ee aa 1f 8a f0 56 c9 02  e1
18a0 : d0 6a a2 00 bd bc 1f 38  fd
18a8 : fd ba 1f b0 10 bd bc 1f  00
18b0 : a8 bd ba 1f 9d bc 1f 98  37
18b8 : 9d ba 1f 90 e7 9d c4 1f  49
18c0 : e8 e0 02 d0 df 20 57 0f  2d
18c8 : a2 00 20 fa 19 8c c2 1f  11
18d0 : 8d c3 1f ae c4 1f ac c5  60
18d8 : 1f a9 bf 91 57 88 10 f9  dc
18e0 : a5 57 18 69 28 85 57 90  91
18e8 : 02 e6 58 ca 10 e8 ad 96  f9
18f0 : 1f 8d ae 1f 60 ad aa 1f  c2
18f8 : c9 03 d0 f8 ce aa 1f ad  b0
1900 : ae 1f 20 64 0e 4c c5 18  5d
1908 : ce aa 1f 60 ad be 1f 18  7d
1910 : 6d c4 1f 8d c0 1f c9 19  b7
1918 : b0 ee ad bf 1f 18 6d c5  97
1920 : 1f 8d c1 1f c9 28 b0 e0  bc
1928 : 20 48 0f a2 04 20 fa 19  e4
1930 : a5 57 38 ed c2 1f a5 58  b9
1938 : ed c3 1f a9 00 2a 2a 85  09
1940 : f9 69 04 aa 20 fa 19 84  8b
1948 : 59 85 5a a6 f9 20 fa 19  8e
1950 : a0 03 06 57 26 58 06 59  ce
1958 : 26 5a 88 d0 f5 ad ae 1f  ad
1960 : a0 06 d9 96 1f f0 03 88  e3
1968 : d0 f8 a5 58 18 88 30 04  b7
1970 : 69 20 90 f9 85 58 a9 00  0e
1978 : a0 08 a6 f9 f0 04 a9 ff  db
1980 : a0 f8 84 5c 85 5d ad c4  cc
1988 : 1f 85 fb 78 a9 34 85 01  cc
1990 : a2 ff a0 07 b1 57 91 59  0a
1998 : 88 10 f9 a0 02 b9 57 00  06
19a0 : 18 65 5c 99 57 00 b9 58  c2
19a8 : 00 65 5d 99 58 00 88 88  9e
19b0 : f0 eb e8 ec c5 1f 90 da  bb
19b8 : e0 27 d0 df c6 fb 10 d0  8a
19c0 : a9 37 85 01 58 ad 89 1f  de
19c8 : 10 2f a2 04 20 fa 19 84  e0
19d0 : 59 85 5a ad ae 1f cd 96  80
19d8 : 1f f0 05 20 b1 0f f0 02  10
19e0 : a9 40 18 6d c3 1f 38 e9  47
19e8 : 04 85 58 ad c2 1f 85 57  64
19f0 : ae c4 1f 20 1f 1a 20 57  be
19f8 : 0f 60 a9 00 85 58 bd ba  29
1a00 : 1f 0a 0a 18 7d ba 1f 0a  e8
1a08 : 0a 26 58 0a 26 58 7d bb  0f
1a10 : 1f 90 02 e6 58 85 57 a8  35
1a18 : a5 58 09 04 85 58 60 ac  a2
1a20 : c5 1f b1 57 91 59 88 10  f2
1a28 : f9 a5 57 18 69 28 85 57  69
1a30 : 90 02 e6 58 a5 59 18 69  de
1a38 : 28 85 59 90 02 e6 5a ca  e2
1a40 : 10 dd 60 20 be 14 a0 03  70
1a48 : 46 58 66 57 88 d0 f9 a5  81
1a50 : 58 49 44 85 5a a5 57 85  4a
1a58 : 59 a0 00 ad 8d 02 4a ad  25
1a60 : 89 1f 29 01 d0 15 b0 06  68
1a68 : ad 20 d0 4c 70 1a b1 59  34
1a70 : 29 0f 85 fd b1 57 29 f0  9e
1a78 : 4c 91 1a b0 0a ad 20 d0  5a
1a80 : 0a 0a 0a 0a 4c 89 1a b1  30
1a88 : 59 29 f0 85 fd b1 57 29  80
1a90 : 0f 05 fd 91 57 60 20 e8  9e
1a98 : 1c a0 04 b9 10 fd d9 e0  57
1aa0 : 3f d0 53 88 10 f5 a9 [illegible]  01
1aa8 : 85 58 a5 5b c9 c0 90 [illegible]  17
1ab0 : a9 bf 29 f0 85 fd 4a 4a  a7
1ab8 : 65 fd 85 fd a5 5d 4a a5  f7
1ac0 : 5c 6a 4a 4a 4a 18 65 fd  24
1ac8 : 0a 90 02 e6 58 85 57 a0  c8
1ad0 : 01 b1 57 aa 88 b1 57 29  9b
1ad8 : 40 d0 1b b1 57 30 18 c9  68
1ae0 : 04 b0 08 8d 81 1f 8a a8  7d
1ae8 : 4c 48 0d 48 8a a8 20 48  a4
1af0 : 0d 68 a8 4c 48 0d 60 a8  a5
1af8 : 29 01 8d 81 1f 08 98 4a  5f
1b00 : 4a a8 28 d0 01 88 8c 20  89
1b08 : d0 8c 27 d0 8a a8 4c 48  b2
1b10 : 0d 48 29 01 d0 1f a9 20  99
1b18 : 85 58 a2 e0 78 a9 34 85  3f
1b20 : 01 a9 00 85 57 a8 91 57  56
1b28 : c8 d0 fb e6 58 ca d0 f6  41
1b30 : a9 37 85 01 58 a2 00 8e  ae
1b38 : 17 d0 8e 1d d0 8e 1c d0  92
1b40 : 8e 1b d0 20 c8 1c a9 04  b0
1b48 : 8d 20 d0 a0 25 b9 91 1b  ca
1b50 : 99 84 1f 88 10 f7 68 8d  82
1b58 : 89 1f 10 23 a9 00 8d 97  d9
1b60 : 1f ad 89 1f 29 01 d0 17  a8
1b68 : a9 40 85 58 ad 88 1f a2  7f
1b70 : 1c a0 00 84 57 91 57 c8  5e
1b78 : d0 fb e6 58 ca d0 f6 ad  75
1b80 : 89 1f 29 40 f0 08 20 f2  a1
1b88 : 10 a9 02 8d ae 1f 4c 74  9d
1b90 : 1c 00 8c ae 00 f6 00 01  5f
1b98 : 01 08 18 a0 01 08 15 60  1d
1ba0 : 20 06 00 01 07 02 03 04  78
1ba8 : 05 06 8c 8c 8c 8c ae ae  aa
1bb0 : ae ae 00 00 00 00 a9 93  83
1bb8 : 20 d2 ff a9 01 8d 15 d0  e9
1bc0 : 8d 10 d0 a9 10 8d 00 d0  ce
1bc8 : a9 88 8d 01 d0 a9 04 85  ae
1bd0 : 58 a9 d8 85 5a a2 00 86  ab
1bd8 : 57 86 59 a9 19 85 fb a0  ed
1be0 : 00 bd c0 02 8d ba 1f bd  d6
1be8 : c1 02 8d bb 1f bd c2 02  74
1bf0 : 8d bc 1f a9 06 91 59 c0  ac
1bf8 : 18 b0 1c a5 fb c9 05 90  67
1c00 : 16 ad 93 1f 49 80 91 57  43
1c08 : 2e bc 1f 2e bb 1f 2e ba  05
1c10 : 1f 90 04 a9 04 91 59 c8  71
1c18 : c0 28 d0 d7 e8 e8 e8 a5  e0
1c20 : 57 18 69 28 85 57 85 59  bf
1c28 : 90 04 e6 58 e6 5a c6 fb  d3
1c30 : d0 ad a9 06 8d 27 d0 a5  a3
1c38 : 5b 8d ba 1f a5 5c 8d bb  d7
1c40 : 1f a2 00 20 fa 19 60 20  ef
1c48 : 57 0f ad 18 d0 29 f7 8d  e6
1c50 : 18 d0 ad 11 d0 29 df 8d  4f
1c58 : 11 d0 20 cc ff a9 93 20  4f
1c60 : d2 ff a9 0f 8d 21 d0 a9  f7
1c68 : 06 8d 86 02 a9 00 85 c6  55
1c70 : 8d 15 d0 60 20 e1 1c a2  8f
1c78 : 00 a0 00 b9 f0 1c 85 57  b4
1c80 : f0 27 c9 ff d0 04 e8 c8  d8
1c88 : d0 f1 b9 f1 1c 9d 40 03  b3
1c90 : e8 b9 f2 1c 9d 40 03 e8  4f
1c98 : b9 f3 1c 9d 40 03 e8 c6  53
1ca0 : 57 d0 e7 c8 c8 c8 c8 d0  0a
1ca8 : d2 ad 18 d0 09 08 8d 18  a8
1cb0 : d0 ad 11 d0 09 20 8d 11  9f
1cb8 : d0 20 48 0f ad 20 d0 8d  c7
1cc0 : 27 d0 ad 84 1f 4c 07 0e  d8
1cc8 : a9 00 a0 3f 99 c0 02 88  3a
1cd0 : 10 fa 60 aa f0 0a a0 00  ad
1cd8 : ea ea 88 d0 fb ca d0 f8  bf
1ce0 : 60 a5 cb c9 40 d0 fa 60  76
1ce8 : 20 a2 12 29 10 d0 f9 60  33
1cf0 : 08 00 20 00 02 00 00 00  21
1cf8 : 01 ff 07 f8 02 00 00 00  fa
1d00 : 08 00 20 00 ff 06 00 00  41
1d08 : 00 01 03 ff 00 08 02 01  94
1d10 : 00 01 03 ff 00 05 00 00  7a
1d18 : 00 ff 01 ff ff ff 13 c0  26
1d20 : 00 03 01 ff ff ff 00 20  22
1d28 : e1 1c a9 20 8d 93 1f a9  cb
1d30 : 00 85 5b 85 5c 8d 83 1f  f9
1d38 : 20 47 1c 20 b6 1b 20 55  76
1d40 : 1d 20 2e 1e 20 7d 1e a9  76
1d48 : 30 20 d3 1c a5 cb c9 3c  59
1d50 : d0 ec 4c 74 1c a5 cb cd  f2
1d58 : 80 1f d0 01 60 8d 80 1f  6f
1d60 : c9 24 d0 25 a0 3c b9 c0  68
1d68 : 02 20 0a 1e 99 bc 1f b9  30
1d70 : c2 02 20 0a 1e 99 ba 1f  55
1d78 : b9 c1 02 20 0a 1e 99 bb  06
1d80 : 1f 88 88 88 10 e0 4c 19  82
1d88 : 1e c9 16 d0 14 a0 00 a2  b6
1d90 : 3e b9 c0 02 20 0a 1e 9d  21
1d98 : ba 1f c8 ca 10 f3 4c 19  71
1da0 : 1e c9 11 d0 44 a9 02 85  a6
1da8 : fb a4 fb 84 fd be 28 1e  37
1db0 : b9 2b 1e 85 fa a9 03 85  4b
1db8 : fc a9 08 85 fe a4 fa 8a  52
1dc0 : 48 a9 00 3e c0 02 6a e8  3c
1dc8 : e8 e8 88 d0 f6 a4 fd 99  20
1dd0 : ba 1f c8 c8 c8 84 fd 68  de
1dd8 : aa c6 fe d0 e0 e8 c6 fc  29
1de0 : d0 d7 c6 fb 10 c3 4c 19  4f
1de8 : 1e c9 1a d0 0b ad 93 1f  36
1df0 : 49 6f 8d 93 1f 4c b6 1b  2c
1df8 : c9 33 d0 0d ad 8d 02 29  d2
1e00 : 01 f0 06 20 c8 1c 4c b6  0b
1e08 : 1b 60 85 fe 84 fd a0 08  5f
1e10 : 26 fe 6a 88 d0 fa a4 fd  d4
1e18 : 60 a0 3e b9 ba 1f 99 c0  1c
1e20 : 02 88 10 f7 20 b6 1b 60  4e
1e28 : 27 0f 00 08 08 05 20 a2  46
1e30 : 12 a0 00 29 0f f0 35 4a  99
1e38 : a6 5b f0 04 90 02 c6 5b  33
1e40 : 4a 90 06 e0 14 b0 02 e6  0d
1e48 : 5b 4a a6 5c f0 04 90 02  73
1e50 : c6 5c 4a 90 06 e0 17 b0  0e
1e58 : 02 e6 5c ad 93 1f 49 80  f2
1e60 : 91 57 20 37 1c ad 93 1f  48
1e68 : a0 00 91 57 ad 83 1f 49  5e
1e70 : 01 8d 83 1f f0 06 b1 57  b1
1e78 : 49 80 91 57 60 ad 82 1f  0c
1e80 : 29 10 f0 3c 20 f7 14 a5  d2
1e88 : 5c 4a 4a 4a a0 03 18 65  33
1e90 : 5b 88 d0 fb a8 a5 5e ae  71
1e98 : 8d 02 d0 0b 19 c0 02 99  8f
1ea0 : c0 02 a9 04 4c b1 1e 49  aa
1ea8 : ff 39 c0 02 99 c0 02 a9  af
1eb0 : 06 a8 a5 58 aa 49 dc 85  f2
1eb8 : 58 98 a0 00 91 57 86 58  23
1ec0 : 60 a0 03 d9 22 1f f0 03  51
1ec8 : 88 d0 f8 c8 98 0a 0a aa  67
1ed0 : a0 03 ca bd 26 1f 99 b6  8b
1ed8 : 1f 88 10 f6 ad 80 1f 4a  0e
1ee0 : 4a 4a 8d b4 1f a9 04 ae  f6
1ee8 : b3 1f ac b4 1f 20 ba ff  ca
1ef0 : a0 02 b1 2d 48 c8 b1 2d  8f
1ef8 : aa c8 b1 2d a8 68 20 bd  e2
1f00 : ff ae b6 1f 86 57 ac b7  2d
1f08 : 1f 84 58 ad b4 1f d0 06  c9
1f10 : a9 00 20 d5 ff 60 a9 57  d4
1f18 : ae b8 1f ac b9 1f 20 d8  46
1f20 : ff 60 47 46 53 5a 00 20  32
1f28 : 00 40 00 20 00 44 c0 02  76
1f30 : ff 02 00 20 00 28 a2 03  06
1f38 : 20 c6 ff 20 cf ff 20 d2  e2
1f40 : ff d0 f8 a9 0d 20 d2 ff  38
1f48 : 20 cc ff 60 00 00 00 00  da
```

Listing 2. HI-EXE (Schluß)

```
1 REM ZEILEN WEGLASSEN !!                        <101>
2 :                                              <234>
3 REM **********************************         <134>
4 REM *                                 *        <053>
5 REM *            HI-EDDI              *        <112>
6 REM *                                 *        <055>
7 REM *   VON:    HANS HABERL           *        <161>
8 REM *           BAHNHOFSTR. 3         *        <146>
9 REM *           8018  GRAFING         *        <238>
10 REM*          TEL. 08092/7152        *        <135>
11 REM*                                 *        <060>
12 REM*********************************          <143>
13 :                                             <245>
50 IF A<2 THEN A=A+2:LOAD"HI-EXE",8,1            <107>
60 IF A=3 THEN 150                               <027>
70 IF A=4 THEN 650                               <200>
100 INPUT"BETRIEBSART";C:POKE 780,C:POKE 5
   6,13:CLR:W$="123456":SYS 3328                 <159>
110 IF PEEK(8073)AND 64 THEN A=3:LOAD"MENU
   E",8,1                                        <087>
150 SYS 3337:A=PEEK(8064)                        <119>
160 ON A+1 GOSUB 200,250,300,290,400,600,2
   90,290,500,510,300:GOTO 150                   <254>
200 OPEN 3,8,0,"$0":GET#3,C$,C$                  <154>
210 GET#3,C$,C$,L$,H$:IF C$=""THEN CLOSE 3
   :WAIT 198,1:RETURN                            <094>
220 PRINT 256*ASC(H$+CHR$(0))+ASC(L$+CHR$(
   0));:SYS 3334:GOTO 210                        <018>
250 PRINT"{CLR,DOWN}SEQUENZ:":PRINT"{DOWN,
   2SPACE}"W$:INPUT"{HOME,3DOWN}";W$:IF L
   EN(W$)<2 THEN 250                             <062>
290 RETURN                                       <094>
300 POKE 8115,8:PRINT"{CLR,DOWN}G=GRAFIKBI
   LD":IF PEEK(8073)>127 THEN PRINT"F=FAR
   BBILD"                                        <036>
310 PRINT"S=SPRITE":PRINT"Z=ZEICHENSATZ":P
   RINT"{DOWN,SPACE}-->";                        <017>
320 GET C$:IF C$<>"G"AND C$<>"F"AND C$<>"S
   "AND C$<>"Z"THEN 320                          <240>
330 PRINT C$:POKE 780,ASC(C$):H$=W$:INPUT"
   {DOWN}FILENAME";W$:SYS 3331:W$=H$:H$="
   "                                             <046>
340 OPEN 1,8,15,H$:INPUT#1,A,H$:PRINT A;H$
   :IF A>0 THEN WAIT 198,1                       <073>
350 CLOSE 1:RETURN                               <223>
400 POKE 631,34:POKE 198,1:H$="":INPUT"{CL
   R,DOWN}";H$:GOTO 340                          <207>
500 A=8075:GOTO 520                              <053>
510 A=8079                                       <140>
520 D=PEEK(8074):C=PEEK(A+D):PRINT"{CLR,DO
   WN}F"2*D+1:PRINT"SCHRITTWEITE:"C,             <055>
530 INPUT C:IF C<1 OR C>160 THEN 530             <066>
540 POKE A+D,C:RETURN                            <189>
600 INPUT"{CLR,DOWN}NUMMER 1.BILD";D:C=(D
   AND 7)                                        <087>
610 INPUT"{DOWN}NUMMER 2.BILD (0=NUR 1 BIL
   D)";D:C=C+8*(D AND 7):IF D>0 THEN C=C+
   64                                            <067>
620 IF D=0 THEN PRINT"{DOWN}GROSS";:GOSUB
   800:IF C$="J"THEN C=C+128                     <075>
630 PRINT"{DOWN}DRUCKEN";:GOSUB 800:IF C$=
   "N"THEN 660                                   <104>
640 IF A=5 THEN A=4:LOAD"HI-PRINT",8,1           <144>
650 POKE 780,C:SYS 3328:PRINT"{DOWN}NOCHMA
   L";:GOSUB 800:IF C$="J"THEN 600               <150>
660 IF A=5 THEN RETURN                           <209>
670 A=1:GOTO 50                                  <194>
800 PRINT" ? (J/N)";                             <205>
810 GET C$:IF C$<>"J"AND C$<>"N"THEN 810         <210>
820 PRINT C$:RETURN                              <230>
```

© 64'er

Listing 3: HI-EDDI Steuerprogramm. Bitte beachten Sie die Eingabehinweise auf Seite 16.


```
10 REM**********************************         <141>
20 REM*                                *         <069>
30 REM*            HI-EDDI             *         <137>
40 REM*        VON HANS HABERL         *         <158>
50 REM*                                *         <099>
60 REM*  DATA'S FUER DIE STEUERZEILE    *         <199>
70 REM*    DER BEISPIEL-MENUETAFEL     *         <145>
80 REM**********************************         <211>
90 REM JE ZWEI AUFEINANDERFOLGENDE DATA-ZE
   ILEN                                          <254>
100 REM ERGEBEN EINE ZEILE IN DER MENUETAF
   EL (=20 FELDER)                               <079>
110 DATA 64,0,64,0,64,0,64,0,64,0,64,0,64,
   0,64,0,64,0,64,0                              <215>
120 DATA 64,0,64,0,64,0,64,0,64,0,64,0,64,
   0,64,0,64,0,64,0                              <225>
130 DATA 0,18,0,18,0,18,0,20,0,20,0,20,0,2
   2,0,22,0,22                                   <097>
140 DATA 0,13,0,13,0,13,0,10,0,10,0,10,0,5
   6,56,23,56,38,56,30,1,51                      <172>
150 DATA 0,18,0,18,0,18,0,20,0,20,0,20,0,2
   2,0,22,0,22                                   <117>
160 DATA 0,13,0,13,0,13,0,10,0,10,0,10,0,5
   9,59,23,59,38,59,30,1,51                      <234>
170 DATA 0,42,0,42,0,42,0,41,0,41,0,41,0,9
   ,0,9,0,9                                      <143>
180 DATA 0,26,0,26,0,26,0,14,0,14,0,14,0,8
   ,8,23,8,38,8,30,1,51                          <058>
190 DATA 0,42,0,42,0,42,0,41,0,41,0,41,0,9
   ,0,9,0,9                                      <163>
200 DATA 0,26,0,26,0,26,0,14,0,14,0,14,0,1
   1,11,23,11,38,11,30,0,33                      <246>
210 DATA 0,17,0,17,0,17,0,36,0,36,0,36,1,9
   ,1,9,1,9                                      <063>
220 DATA 1,60,1,60,1,60,1,60,1,60,1,60,0,1
   6,16,23,16,38,16,30,0,33                      <070>
230 DATA 0,17,0,17,0,17,0,36,0,36,0,36,1,9
   ,1,9,1,9                                      <083>
240 DATA 1,60,1,60,1,60,1,60,1,60,1,60,0,1
   9,19,23,19,38,19,30,0,33                      <132>
250 DATA 128,21,132,21,136,21,140,21,144,2
   1,148,21,152,21,156,21,160,21                 <082>
260 DATA 164,21,168,21,172,21,176,21,180,2
   1,184,21,188,21,0,4,0,5,0,6,0,3               <097>
270 DATA 128,28,132,28,136,28,140,28,144,2
   8,148,28,152,28,156,28,160,28                 <111>
280 DATA 164,28,168,28,172,28,176,28,180,2
   8,184,28,188,28,1,4,1,5,1,6,1,3               <137>
290 DATA 129,21,133,21,137,21,141,21,145,2
   1,149,21,153,21,157,21,161,21                 <155>
300 DATA 165,21,169,21,173,21,177,21,181,2
   1,185,21,189,21,2,4,2,5,2,6,2,3               <140>
310 DATA 129,28,133,28,137,28,141,28,145,2
   8,149,28,153,28,157,28,161,28                 <182>
320 DATA 165,28,169,28,173,28,177,28,181,2
   8,185,28,189,28,0,29,0,29,0,31,0,31           <088>
330 DATA 2,18,2,18,2,18,2,42,2,42,2,4
   2,2,42,2,13,2,13,2,13,2,13                    <192>
340 DATA 2,20,2,20,2,20,2,20,2,41,2,41,2,4
   1,2,41                                        <211>
350 REM  KENNUNG "CBM80" ,ALS MENUEKENNUNG
   MISSBRAUCHT                                   <072>
360 DATA 195,194,205,56,48                       <044>
370 :                                            <092>
380 POKE 56,32:CLR:REM IM SPEICHER BEFINDL
   ICHE MENUETAFEL SCHUETZEN                     <226>
390 FOR I=15872 TO 15872+484:READ A:POKE I
   ,A:S=S+A:NEXT I                               <028>
400 IF S<>18329 THEN PRINT"DATA-FEHLER!"         <099>
```

© 64'er

Listing 4: Ein Beispiel für den Aufbau einer Steuertafel für ein selbsterstelltes Menü. Bitte beachten Sie die Eingabehinweise auf Seite 16.

Vier Pseudo-VICs mit 32 Sprites

Sie wollen mit 32 Sprites und vier Bildschirmbereichen gleichzeitig arbeiten? Nichts leichter als das. Mit Provic 64 können simultan hochauflösende Grafik, Text oder veränderte Zeichensätze dargestellt werden.

Wir (Jürgen, 21, Physikstudent, und Stefan, 18, Schüler) haben uns Mitte 1983 einen Commodore 64 angeschafft. Schon nach kurzer Zeit stellte sich der übliche Frust über die schlechte Dokumentation und die schwierige Informationsbeschaffung ein, besonders wenn es um die speziellen Grafikfähigkeiten dieses Computers geht. So saßen wir oft stundenlang vor dem Bildschirm, der nur undefinierbare Zeichen zeigte, weil wir bei dem Versuch, die Geheimnisse des C 64 zu enträtseln, in irgendeinen unbekannten Darstellungsmodus geraten waren.

Dabei entdeckten wir, daß der C 64 nicht nur acht, sondern auch 16, 24, 32 oder noch mehr Sprites gleichzeitig zeigen kann. Zusätzlich ergibt sich die Möglichkeit, mehrere Bildschirmmodi zu mischen.

Nun haben wir uns entschlossen, den Kunstgriff, der dies ermöglicht, anderen C 64-Fans nicht vorzuenthalten. Also entwickelten wir ein Programm in Maschinensprache und dazu ein kleines Demonstrationsprogramm in Basic.

Zum Programm

Durch Aufruf der Initialisierungsroutine wird der Interruptmechanismus des C 64 verändert. Nicht mehr der Timer der CIA 1, sondern der VIC 6567 löst jetzt den Interrupt aus, und zwar synchron zum Takt des Bildschirmsignals. Außerdem werden vier sogenannte Pseudo-VICs eingerichtet. Alle POKEs, von Spritebewegung über Bildschirmfarbe bis zur Grafikkonfiguration, müssen ab jetzt in diese Pseudo-VICs erfolgen. Jeder dieser Pseudo-VICs ist für einen der vier Bildschirmbereiche zuständig:

Der Bildschirm wird in vier waagerechte Bereiche aufgeteilt, deren Grenzlinien fast beliebig nach oben oder unten verschoben werden können. Jeder einzelne Bereich kann acht Sprites darstellen und eine eigene Farb- und Grafikkonfiguration aufweisen. So können zum Beispiel Normalschrift, HiRes-Grafik, Multicolor-Grafik und eventuell ein selbstdefinierter Zeichensatz gleichzeitig auf dem Bildschirm gezeigt werden.

Provic 64 kann selbstverständlich wieder abgeschaltet werden (bei Kassetten- oder Diskettenoperationen nötig).

Für Maschinensprachefreaks nun eine kurze Funktionsbeschreibung der Interruptroutine:

Bei Aufruf der Einschaltroutine (SYS 52544) wird der IRQ-Vektor auf die Hauptroutine des Provic 64 gestellt und der bisherige Interrupt durch den Timer A der CIA 1 verboten, während der Raster-IRQ des VIC 6567 erlaubt wird.

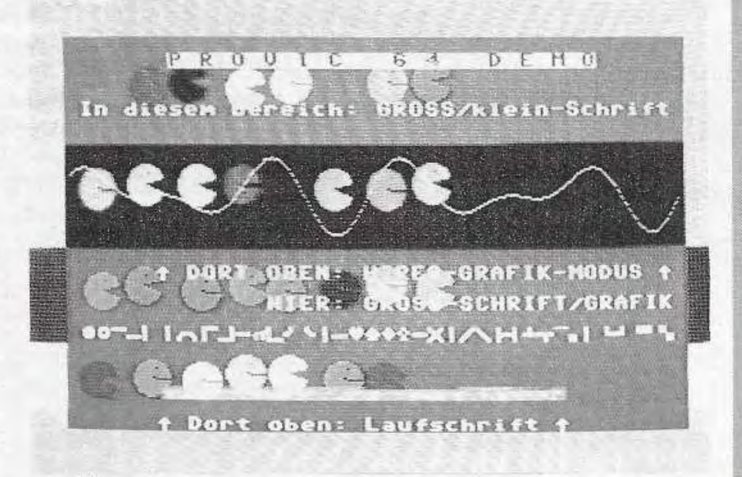
Sobald der Bildschirmstrahl die eingestellte Rasterzeile erreicht, wird ein Interrupt ausgelöst und der Prozessor bearbeitet die Hauptroutine des Provic 64. In dieser wird zunächst anhand eines Zählers ($ CFFF) festgestellt, welcher Bildschirmbereich an der Reihe ist. Dann wird die Rasterzeile, die das Ende dieses Bildschirms kennzeichnet, eingestellt.

Anschließend werden, falls ein entsprechendes Flag gesetzt ist, die Sprite- und andere Bildschirmparameter in den VIC 6567 übertragen. Nach dem Weiterzählen des IR-Zählers ($ CFFF) wird entweder der Interrupt beendet, oder zur IRQ-Routine des Betriebssystems gesprungen (nach jedem vierten Interrupt). So werden in der Sekunde 200 Interrupts (vier pro Fersehbild) ausgelöst und 50 mal in der Sekunde (einmal pro Bild) die normale IRQ-Routine abgearbeitet. Dadurch zählt die interne Uhr TI in 50stel Sekunden und TI$ wird unbrauchbar.

Beim Aufruf der Ausschaltroutine wird der Raster-IRQ des VIC 6567 unterbunden, der Interrupt des Timers A in CIA 1 erlaubt und der IRQ-Vektor auf die IRQ-Routine des Betriebssystems eingestellt.

Handhabung der Pseudo-VICs

Im Grunde ist jeder der vier Pseudo-VICs wie der echte VIC zu behandeln. Ausnahmen sind hier nur die Register 30 (Sprite-Sprite-Kollision) und 31 (Sprite-Hintergrund-Kollision), die sich auf den jeweils vorausgegangenen Bildschirmbereich beziehen. Die Register 19 und 20 (Lightpenkoordinaten), sowie 25 und 26 (IRQ-Flags und -maske) werden nicht behandelt, da diese Funktionen nur direkt über den VIC 6567 sinnvoll gehandhabt werden können. Außerdem hat jeder Pseudo-VIC noch zusätzliche Register für zwei Flags (REG 47 und REG 57), acht Sprite-Pointer (REG 48 bis REG 55),



So werden die Fähigkeiten von »Provic 64« demonstriert

Die vier Basisadressen der PVICs sind:

PVIC 1	52992 ($ CF00)	= REG 0
PVIC 2	53056 ($ CF40)	= REG 0
PVIC 3	53120 ($ CF80)	= REG 0
PVIC 4	53184 ($ CFC0)	= REG 0

Da die Pseudo-VICs praktisch gleichberechtigt sind, hier die Registerbeschreibung eines Pseudo-VICs:

REG 0:	X-Koordinate des Sprite 0
REG 1:	Y-Koordinate des Sprite 0 Beachte: Die Y-Koordinaten sollten im Bereich des zugehörigen Bildschirmbereichs liegen, sonst ist der Sprite nicht zu sehen. Näheres siehe unten.
REG 2 bis 15:	Wie REG 0 und 1, aber für Sprites 1 — 7
REG 16:	MSB (höchstes Bit) der X-Koordinaten
REG 17:	Bits 0 bis 2: Y-Abstand der Zeichen vom oberen Bildrand in Rasterzeilen (Softscrolling!)
REG 17:	Bit 3: Umschaltung 24/25-Zeilendarstellung Bit 4: Bild an/aus: es hat keinen Sinn, das Bild teilweise ausschalten zu wollen, da der VIC dieses Bit nur einmal pro Fernsehbild prüft (also entweder das ganze Bild an oder aus) Bit 5: HiRes-Grafik-Modus an Bit 6: Hintergrundmehrfarb-Modus an Bit 7: Nummer der Interrupt-Rasterzeile Bit 8; es hat wenig Sinn, dieses Bit zu setzen, da so nur Rasterzeilen angesprochen werden, die unterhalb des Bildschirms liegen. Ist in irgendeinem Pseudo-VIC die 9-Bit-Zahl für die Rasterzeile größer als 311, so wird überhaupt kein IRQ mehr ausgelöst.
REG 18:	Nummer der Rasterzeile Bits 0 — 7; hier ist anzugeben, wann der nächste Interrupt ausgelöst werden soll, das heißt wo der Bildschirmbereich dieses PVICs zu Ende sein soll. Dabei sollte folgende Reihenfolge eingehalten werden: REG 18: PVIC 1 < PVIC 2 < PVIC 3 <PVIC 4 (Zyklische Vertauschungen möglich!)
REG 19 und 20:	nicht verwendet (siehe oben)
REG 21:	Sprite enable (einschalten)
REG 22:	Bits 0 bis 2: Softscrolling in X-Richtung Bit 3: Umschaltungen 38/40-Spaltendarstellung Bit 4: Multicolor-Modus ein Bit 5 bis 7: unbenutzt
REG 23:	Sprite vergrößern in Y-Richtung
REG 24:	Bits 1 bis 3: Adresse Zeichengenerator (Bits 11 bis 13) Bits 4 bis 7: Adresse Video-RAM (Bits 10 bis 13)
REG 25 und 26:	nicht verwendet (siehe oben)
REG 27:	Sprite-Priorität vor Hintergrund
REG 28:	Flags für Multicolor-Sprites
REG 29:	Sprite vergrößern in X-Richtung
REG 30:	Sprite-Sprite-Kollision
REG 31:	Sprite-Hintergrund-Kollision Achtung: Geben die Kollisionen des vorangegangenen Bildschirmbereichs an: Findet im Bereich von PVIC 3 eine Kollision statt, wird dies im PVIC 4 registriert. Kollisionen im Bereich von PVIC 4 werden im PVIC 1 registriert. Dieses Register muß gelöscht werden, um neue Kollisionen anzeigen zu können!
REG 32:	Rahmenfarbe
REG 33 bis 36:	Hintergrundfarben 0 bis 3
REG 37 und 38:	Multicolor-Sprite-Farben 0 und 1
REG 39 bis 46:	Farben für Sprites 0 bis 7
REG 47:	Flag für Spritebehandlung; nur wenn der Inhalt dieses Registers nicht Null ist, werden die Register, die etwas mit Sprites zu tun haben, vom PVIC in den VIC 6567 übertragen. Das sind REG 0 bis REG 16, REG 21, REG 23, REG 27 bis REG 31, REG 37 bis 46 sowie REG 48 bis 55. Ist der Inhalt Null, gelten für die Sprites die Werte des vorherigen PVICs, während die Kollisionen erst im nächsten PVIC, in dem REG 47 ungleich Null ist, angezeigt werden.
REG 48 bis 55:	Sprite-Pointer für Sprites 0 bis 7; Die Pointer auf die Bitmuster der Sprites werden nicht mehr in die Speicherzellen 2040 bis 2047 geschrieben, sondern in diese Register des PVICs.
REG 56:	In diesem Register muß das Highbyte der Video-RAM-Anfangsadresse plus 3 stehen; normalerweise also 4 + 3 = 7 (da der Bildschirm nach dem Einschalten des Computers bei 1024 beginnt, 1024 = $ 0400). Bei Verlegung des Video-RAMs ist also der Inhalt dieses Registers zu korrigieren.
REG 57:	Flag für Bildschirmparameter-Behandlung; nur wenn der Inhalt dieses Registers nicht Null ist, werden die REG 17, 22, 24, sowie 32 bis 36 und REG 58 in den VIC 6567 übertragen.
REG 58:	Bits 0 und 1: Adreßbist 14 und 15 des VIC 6557; werden nach CIA 2 REG 0 Bits 0 und 1 übertragen. Mit diesen Bits kann Video-RAM, Charaktergenerator, Grafik-Bitmap in 16-KByte-Schritten verschoben werden. Da die Bits low-aktiv sind, sind sie beim Einschalten gesetzt (also REG 58 = 3). Bits 2 bis 7: unbenutzt, immer 0.

Videomatrix-Anfangsadresse Highbyte (REG 56) und die CIA 2, REG 0, Bits 0 und 1 (REG 58) (Adreßbits 14 und 15 des VIC 6567).

Übergang eines Sprites zwischen zwei Bildschirmbereichen:

Soll ein Sprite zwischen zwei Bildschirmbereichen wechseln, muß in beiden Bereichen derselbe Sprite (also zum Beispiel beidesmal Sprite 4) die gleiche Position besitzen, und zwar so lange, wie der Sprite die Trennlinie zwischen den Bereichen überdeckt. Wird dies nicht beachtet, werden die entsprechenden Sprites zerschnitten und verschoben.

Aktivieren von Provic 64: Von Basic aus mit SYS 52544 und von Maschinensprache aus mit JSR $CD40.

Ausschalten von Provic 64: Von Basic aus mit SYS 52970 und von Maschinensprache aus mit JSR $CEEA.

Eingabe von Provic 64:

Die Eingabe des Listing 1 erfolgt mit dem MSE, das Speichern ebenfalls.

Laden von Provic 64:

Im Programm am besten mit der Zeile IF PEEK (52544) > < 120 THEN LOAD "PROVIC 64", Gerätenummer, 1 die am Anfag des Basic-Programms stehen sollte.

Das Demonstrationsprogramm (Listing 2) zeigt einige der Vorzüge von Provic 64. Bis der Bildschirmaufbau fertig ist, müssen Sie eine Weile warten. Es ist nur als Anregung gedacht, deshalb verzichten wir hier auf eine nähere Beschreibung.

Bei Kassetten- und Disketten- sowie Drucker-Zugriffen muß Provic vorher abgeschaltet werden. Sonst beginnt der Bildschirm wild zu flackern und Provic stürzt vielleicht ab.

Provic 64 ist nicht nur für Basic-Programmierer, sondern vor allem auch für Maschinensprache-Freaks gedacht, da erst durch schnelle Maschinenprogramme die Möglichkeiten von Provic 64 voll ausgeschöpft werden können.

(Jürgen und Stefan Haas/bs)

Da die Pseudo-VICs praktisch gleichberechtigt sind, hier die Registerbeschreibung eines Pseudo-VICs:

Belegter Adreßraum:

$CD40	Einschaltroutine
$CD58	Interruptroutine
$CEEA	Ausschaltroutine
$CF00	Pseudo-VIC 1
$CF40	Pseudo-VIC 2
$CF80	Pseudo-VIC 3
$CFC0	Pseudo-VIC 4
$CFFF	Interruptzähler
Provic 64 einschalten:	in Basic: SYS 52544 in Maschinensprache: JSR $CD40
Provic 64 ausschalten:	in Basic: SYS 52970 in Maschinensprache JSR $CEEA
Benutzte RAM-Adressen:	
in der Zero-Page:	187 ($BB) 188 ($BC)

Rechenzeitzuwachs bei aktiviertem Provic 64:
alle Spriteflags (REG 47) und Bildschirmparameterflags (REG 57) gelöscht: + 2,5 %
für jedes gesetzte Spriteflag (REG 47): zirka + 2,4 %
für jedes gesetzte Bildschirmflag (REG 57): zirka + 0,5 %
alle Sprite- und Bildschirmflags gesetzt: zirka + 15,0 %

Falls der C 64 abstürzt, rettet Run-Stop/Restore!
Die Zeitvariable TI zählt bei aktiviertem Provic 64 in 50stel Sekunden (statt 60stel);
TI$ wird somit unbrauchbar.
Zeiger für Interruptaussprung von PVIC 3 bis PVIC 4: $CEE5
Zeiger für Interruptaussprung von PVIC 1: $CEE8

Tabellarische Übersicht zu Provic 64

```
programm : provic 64      cd40 d000
-----------------------------------
cd40 : 78 a9 58 a2 cd 8d 14 03   97
cd48 : 8e 15 03 a9 01 8d 0d dc   c1
cd50 : 8d 1a d0 8d ff cf 58 60   71
cd58 : a9 01 8d 19 d0 ac ff cf   1a
cd60 : f0 0f 88 f0 09 88 f0 03   b7
cd68 : a2 c0 2c a2 80 2c a2 40   3e
cd70 : 2c a2 00 bd 12 cf 8d 12   9f
cd78 : d0 bd 2f cf d0 03 4c 8d   5e
cd80 : ce ad 1e d0 1d 1e cf 9d   04
cd88 : 1e cf ad 1f d0 1d 1f cf   ef
cd90 : 9d 1f cf bd 15 cf 8d 15   99
cd98 : d0 bd 17 cf 8d 17 d0 bd   57
cda0 : 1d cf 8d 1d d0 bd 00 cf   46
cda8 : 8d 00 d0 bd 01 cf 8d 01   e8
cdb0 : d0 bd 02 cf 8d 02 d0 bd   81
cdb8 : 03 cf 8d 03 d0 bd 04 cf   11
cdc0 : 8d 04 d0 bd 05 cf 8d 05   4a
cdc8 : d0 bd 06 cf 8d 06 d0 bd   ba
cdd0 : 07 cf 8d 07 d0 bd 08 cf   be
cdd8 : 8d 08 d0 bd 09 cf 8d 09   ac
cde0 : d0 bd 0a cf 8d 0a d0 bd   f3
cde8 : 0b cf 8d 0b d0 bd 0c cf   6a
cdf0 : 8d 0c d0 bd 0d cf 8d 0d   0f
cdf8 : d0 bd 0e cf 8d 0e d0 bd   2c
ce00 : 0f cf 8d 0f d0 bd 10 cf   17
ce08 : 8d 10 d0 bd 1b cf 8d 1b   26
ce10 : d0 bd 1c cf 8d 1c d0 bd   38
ce18 : 25 cf 8d 25 d0 bd 26 cf   60
ce20 : 8d 26 d0 bd 27 cf 8d 27   22
ce28 : d0 bd 28 cf 8d 28 d0 bd   b4
ce30 : 29 cf 8d 29 d0 bd 2a cf   0d
ce38 : 8d 2a d0 bd 2b cf 8d 2b   84
ce40 : d0 bd 2c cf 8d 2c d0 bd   ed
ce48 : 2d cf 8d 2d d0 bd 2e cf   b9
ce50 : 8d 2e d0 bd 38 cf 85 bc   72
ce58 : a9 f8 85 bb a0 00 bd 30   b7
ce60 : cf 91 bb c8 bd 31 cf 91   c7
ce68 : bb c8 bd 32 cf 91 bb c8   47
ce70 : bd 33 cf 91 bb c8 bd 34   4e
ce78 : cf 91 bb c8 bd 35 cf 91   00
ce80 : bb c8 bd 36 cf 91 bb c8   df
ce88 : bd 37 cf 91 bb bd 39 cf   35
ce90 : f0 43 bd 3a cf 29 03 9d   66
ce98 : 3a cf ad 00 dd 29 fc 1d   7b
cea0 : 3a cf 8d 00 dd bd 11 cf   d5
cea8 : 8d 11 d0 bd 16 cf 8d 16   ec
ceb0 : d0 bd 18 cf 8d 18 d0 bd   b7
ceb8 : 20 cf 8d 20 d0 bd 21 cf   46
cec0 : 8d 21 d0 bd 22 cf 8d 22   e5
cec8 : d0 bd 23 cf 8d 23 d0 bd   ea
ced0 : 24 cf 8d 24 d0 e0 c0 d0   7c
ced8 : 05 a9 ff 8d ff cf ee ff   9d
cee0 : cf 8a f0 03 4c 81 ea 4c   a6
cee8 : 31 ea 78 a9 31 a2 ea 8d   d1
cef0 : 14 03 8e 15 03 a9 81 8d   6b
cef8 : 0d dc a9 00 4c 50 cd ea   32
cf00 : 00 00 00 00 00 00 00 00   01
cf08 : 00 00 00 00 00 00 00 00   09
cf10 : 00 1b 5f 00 00 00 c8 00   99
cf18 : 15 79 f0 00 00 00 00 00   26
cf20 : 0e 06 01 02 03 04 00 01   04
cf28 : 02 03 04 05 06 07 08 00   06
cf30 : 00 00 00 00 00 00 00 00   31
cf38 : 07 01 03 00 00 00 00 00   81
cf40 : 00 00 00 00 00 00 00 00   41
cf48 : 00 00 00 00 00 00 00 00   49
cf50 : 00 1b 91 00 00 00 c8 00   65
cf58 : 15 79 f0 00 00 00 00 00   66
cf60 : 0e 06 01 02 03 04 00 01   44
cf68 : 02 03 04 05 06 07 08 00   46
cf70 : 00 00 00 00 00 00 00 00   71
cf78 : 07 01 03 00 00 00 00 00   c1
cf80 : 00 00 00 00 00 00 00 00   81
cf88 : 00 00 00 00 00 00 00 00   89
cf90 : 00 1b c3 00 00 00 c8 00   32
cf98 : 15 79 f0 00 00 00 00 00   a6
```

Listing 1. Das Listing »Provic 64«. Bitte beachten Sie die Eingabehinweise auf Seite 18.

Bitte beachten Sie die Eingabehinweise auf Seite 18.

cfa0	:	0e	06	01	02	03	04	00	01	84
cfa8	:	02	03	04	05	06	07	08	00	86
cfb0	:	00	00	00	00	00	00	00	00	b1
cfb8	:	07	01	03	00	00	00	00	00	01
cfc0	:	00	00	00	00	00	00	00	00	c1
cfc8	:	00	00	00	00	00	00	00	00	c9
cfd0	:	00	1b	f5	00	00	00	c8	00	fe
cfd8	:	15	79	f0	00	00	00	00	00	e6
cfe0	:	0e	06	01	02	03	04	00	01	c4
cfe8	:	02	03	04	05	06	07	08	00	c6
cff0	:	00	00	00	00	00	00	00	00	f1
cff8	:	07	01	03	00	00	00	00	00	41

Listing 1. Provic 64 (Schluß)

```
6 REM  DIESES KURZE DEMO-PROGRAMM SOLL      <075>
8 REM  EIN PAAR DARSTELLUNGSFORMEN          <251>
10 REM ZEIGEN, WIE SIE MIT PROVIC-64        <024>
12 REM RELATIV EINFACH ERREICHBAR SIND.     <110>
14 REM        1984 BY GEBR. HAAS            <046>
19 REM                                      <081>
20 REM LADEN DER PROVIC-64 ROUTINE          <069>
21 REM                                      <083>
30 IF PEEK(52544)=120 THEN 100              <014>
40 PRINT"{2DOWN,SPACE}AUF WELCHEM DATENTRA
   EGER IST PROVIC-64"                      <056>
50 PRINT"{DOWN,SPACE}VERFUEGBAR ( FLOPPY =
   8 / DATASETTE =1 )"                      <241>
60 INPUT" ";A$:A=VAL(A$)                    <073>
70 IF A<>1 AND A<>8 THEN 40                 <077>
80 LOAD"PROVIC 64",A,1                      <130>
99 REM                                      <161>
100 REM PSEUDO-VIC'S INITIALISIEREN         <145>
101 REM                                     <163>
110 P1=52992:P2=53056:P3=53120:P4=53184:RE
    M BASISADRESSEN DER PSEUDO-VIC'S        <206>
120 POKE P1+21,255:POKE P1+24,22:POKE P1+4
    7,1:POKE P1+27,255:POKE P2+21,255       <172>
130 POKE P2+17,59:POKE P2+24,24:POKE P2+32
    ,7:POKE P2+47,1:POKE P2+27,255          <071>
140 POKE P3+21,255:POKE P3+32,9:POKE P3+47
    ,1:POKE P3+27,255:POKE P4+32,5          <132>
150 POKE P4+21,255:POKE P4+24,22:POKE P4+4
    7,1:POKE P4+18,230:POKE P4+27,255       <217>
199 REM                                     <005>
200 REM SPRITE-DATEN UEBERTRAGEN            <034>
201 REM                                     <007>
210 RESTORE                                 <004>
220 FOR A=0 TO 126                          <124>
230 READ WERT                               <113>
240 POKE 832+A,WERT                         <152>
250 NEXT A                                  <014>
299 REM                                     <107>
300 REM BILSCHIRM AUFBAUEN                  <127>
301 REM                                     <109>
310 PRINT"{CLR,WHITE,DOWN,RVSON,7RIGHT}P{S
    HIFT-SPACE}R{SHIFT-SPACE}O{SHIFT-SPACE
    }V{SHIFT-SPACE}I{SHIFT-SPACE}C{3SHIFT-
    SPACE}6{SHIFT-SPACE}4{3SHIFT-SPACE}D{S
    HIFT-SPACE}E{SHIFT-SPACE}M{SHIFT-SPACE
    }O"                                     <192>
320 PRINT"{2DOWN,SPACE}IN DIESEM BEREICH:
    GROSS/KLEIN-SCHRIFT{DOWN}":             <044>
330 FOR A=0 TO 7:PRINT"{BLUE}RRRRRRRRRRRRR
    RRRRRRRRRRRRRRRRRRRRRRRRRRR{WHITE}";:N
    EXT A                                   <178>
335 FOR A=0 TO 14:POKE 1289+A,34:NEXT A     <248>
340 PRINT"{6SPACE}↑ DORT OBEN: HIRES-GRAFI
    K-MODUS ↑"                              <024>
350 PRINT"{DOWN,13SPACE}HIER: GROSS-SCHRIF
    T/GRAFIK"                               <093>
360 PRINT"{DOWN,SPACE}QWERTYUIOP*↑LKJHFSAZ
    XCVBNMQWERTFGZXCVB"                     <037>
370 PRINT"{5DOWN,6SPACE}↑ DORT OBEN: LAUFS
    CHRIFT ↑{3UP}":PRINT TAB(6);            <055>
390 SYS 52544:REM PROVIC-64 ACTIVIEREN      <080>
399 REM                                     <207>
400 REM HIRES-GRAFIK ZEICHNEN               <201>
401 REM                                     <209>
410 FOR A=10110 TO 12360:POKE A,0:NEXT A    <212>
420 FOR A=0 TO 8*πSTEP.08                   <248>
430 X=3+A/.08:Y=77-11*SIN(A)-9*COS(A/.7)    <121>
440 AV=8192+320*INT(Y/8)+(Y AND 7)+8*INT(X
    /8)                                     <172>
450 POKE AV,PEEK(AV) OR 2↑(7-(X AND 7))     <072>
460 NEXT A                                  <226>
470 LA$="*** VON HAASISOFT ***{2SPACE}FUER
     DAS 64'ER MAGAZIN{3SPACE}"             <128>
480 LA$=LA$+"***{3SPACE}P{SHIFT-SPACE}R{SH
    IFT-SPACE}O{SHIFT-SPACE}V{SHIFT-SPACE}
    I{SHIFT-SPACE}C{3SHIFT-SPACE}6 4{3SPAC
    E}"                                     <187>
490 LA$=LA$+LEFT$(LA$,25):R=53266           <066>
499 REM                                     <051>
500 REM DEMONSTRATINS-SCHLEIFE              <141>
501 REM                                     <053>
510 REM SPRITES SETZEN                      <013>
511 REM                                     <063>
520 FOR A=0 TO 7                            <017>
530 POKE P1+2*A,30+24*A+7*RND(1):POKE P1+2
    *A+1,60+6*RND(1)                        <158>
540 POKE P1+39+A,RND(1)*16:POKE P1+48+A,13
    .5+RND(1)                               <110>
550 POKE P2+2*A,30+24*A+7*RND(1):POKE P2+2
    *A+1,110+6*RND(1)                       <011>
560 POKE P2+39+A,RND(1)*16:POKE P2+48+A,13
    .5+RND(1)                               <178>
570 POKE P3+2*A,30+24*A+7*RND(1):POKE P3+2
    *A+1,160+6*RND(1)                       <135>
580 POKE P3+39+A,RND(1)*16:POKE P3+48+A,13
    .5+RND(1)                               <246>
590 POKE P4+2*A,30+24*A+7*RND(1):POKE P4+2
    *A+1,207+6*RND(1)                       <059>
600 POKE P4+39+A,RND(1)*16:POKE P4+48+A,13
    .5+RND(1)                               <058>
610 NEXT A                                  <122>
619 REM                                     <173>
620 REM LAUFSCHRIFT SETZEN                  <097>
621 REM                                     <175>
625 FOR LP=1 TO LEN(LA$)-25                 <195>
630 LZ=LZ-1:IF LZ>0 THEN POKE P4+22,LZ OR
    8:FOR A=0 TO 9:NEXT A:GOTO 630          <110>
640 PRINT TAB(6);:WAIT 53265,128:WAIT 5326
    6,64:POKE 53206,15:PRINT LF$:PRINT"{2U
    P}"                                     <193>
660 LZ=7:LF$=MID$(LA$,LP,25)                <244>
670 NEXT LP                                 <019>
680 GET A$:IF A$=""THEN 500                 <173>
690 SYS 52970:REM PROVIC-64 DESAKTIVIEREN   <194>
999 REM                                     <043>
1000 REM SPRITE-DATEN                       <203>
1001 REM                                    <045>
1002 DATA 0,0,0,0,126,0,1,255,128,7,255,22
     4,15,255,240,15,253,240,31,255,248     <077>
1003 DATA 31,255,248,63,255,252,63,255,252
     ,63,243,252,63,252,0,63,255,252,63     <031>
1004 DATA 255,252,31,255,248,31,255,248,15
     ,255,240,15,255,240,7,255,224,1,255    <199>
1005 DATA 128,0,126,0,0,0,0,0,0,126,0,1,25
     5,128,7,255,224,15,255,240,15,251      <154>
1006 DATA 240,31,255,248,31,255,248,63,255
     ,240,63,255,0,63,240,0,63,252,0,63     <003>
1007 DATA 255,0,63,255,224,31,255,248,31,2
     55,248,15,255,240,15,255,240,7,255     <050>
1008 DATA 224,1,255,128,0,126,0             <079>
```


Listing 2. Ein Demonstrations-Programm zu »Provic 64«. Bitte beachten Sie die Eingabehinweise auf Seite 16.

Turtle-Grafik: Die schnelle Schildkröte

Turtle-Grafik ist eine Spracherweiterung, die es in sich hat. Vollständig in Maschinensprache geschrieben, stellt sie einige Befehle zur Verfügung, mit denen Sie komfortabel sehr schnell Bilder erzeugen können.

Wenn Sie dieses Programm (Listing 1) abtippen, werden Sie keinen Ärger mit den vielen DATAs bekommen. Haben Sie einen falschen Wert eingegeben, weist Sie die eingebaute Prüfsummenroutine auf die fehlerhafte Zeile hin und listet sie am Bildschirm. Bitte halten Sie sich daher an die Zeilennumerierung. Nach der Eingabe sollten Sie das Programm speichern und erst dann starten. Wenn alles ok ist, empfehle ich Ihnen, das Demoprogramm (Listing 2) einzugeben. Es vermittelt Ihnen einen sehr guten Eindruck von den Fähigkeiten der Grafikerweiterung. Doch nun zur Turtle-Grafik selbst.

Wird das Programm gestartet, meldet es sich mit: TURTLE GRAPHICS BY PETER MENKE 38911 BASIC BYTES FREE

Nun sind alle Funktionen und Befehle des Programms fest in Basic eingebettet und bleiben bis zum Ausschalten erhalten.

Das Programm unterstützt die Programmierung zweier voneinander völlig unabhängiger Bildschirme:

1. Den normalen Textbildschirm
2. Den Grafikbildschirm, auf dem die hochauflösende Grafik erscheint.

Zwischen den beiden Bildschirmen können Sie mit der Funktionstaste F1 hin- und herschalten. Der neue Befehl HIRES1 schaltet den Grafik-Bildschirm ein (HIRES 0 = ausschalten).

Im folgenden die Befehle im einzelnen:

HIRES1 hi, ra
hi=Hintergrundfarbe (0-15), ra=Randfarbe (0-15)
Die Angaben für Hintergrund und/oder Randfarbe sind nicht unbedingt notwendig, die alten Farben werden dann beibehalten.

CLEAR
Dieser Befehl löscht den gesamten Grafikbildschirm.

REVERS
Dieser Befehl invertiert den gesamten Grafikbildschirm.

COLOR pu, hi, ra
Die Farben der hochauflösenden Grafik werden neu definiert (pu=Punktfarbe). Wie beim HIRES-Befehl können auch hier die Angaben für hi und/oder ra entfallen.

GSAVE »Name«, Gerätenummer
Speichert eine erzeugte Grafik ab.

GLOAD »Name«, Gerätenummer
Lädt eine vorher gespeicherte Grafik in den Computer.

Wenn Sie den Grafikbildschirm einschalten, sehen Sie in der Mitte des Bildschirms einen blinkenden Punkt: den Grafikcursor (oder »Turtle«, zu deutsch »Schildkröte«).

Durch Bewegen dieses Grafikcursors können Sie Linien auf dem Bildschirm zeichnen.

DEG wi
Bestimmt die Bewegungsrichtung der Turtle. Acht Richtungen sind möglich (»wi« kann Werte zwischen 0 und 7 annehmen).

```
    2
  3   1
4   *   0
  5   7
    6
```

MOVE x
Move bewegt die Schildkröte um x Punkte.

Der HIRES-Befehl positioniert den Grafikcursor automatisch auf die Bildschirmmitte. Der eingestellte Winkel ist 0. Außerdem wird Modus 0 eingeschaltet (siehe MODE).

LTURN x (x max.=255)
Dreht den Grafikcursor um x Einheiten nach links.

RTURN x (x max.=255)
Dreht den Grafikcursor um x Einheiten nach rechts.

PLOT x-cor, y-cor
Setzt den Grafikcursor auf eine bestimmte Bildschirmposition. Die obere linke Ecke des Bildschirms hat die Koordinaten 0,0; die rechte untere 319,199.

MODE m (m max 4)
Bei den Befehlen MOVE und PLOT kennt das Programm 4 Modi.
0 = Punkt setzen
1 = Punkt löschen
2 = Punkt invertieren
3 = nichts verändern
Normalerweise ist Modus 0 eingeschaltet. Mit dem MODE-Befehl läßt sich dies ändern.

Gleichzeitige Darstellung von Text und Grafik

Das Programm teilt den Bildschirm in einen Text- und in einen Grafikteil.

Betätigt man bei eingeschaltetem Grafikbildschirm die F3-Taste, so wird im unteren Teil des Bildschirms der untere Teil des normalen Textbildschirms eingeblendet.

Bei nochmaligem Betätigen dieser Taste wird das »Textfenster« wieder ausgeblendet.

WINDOW 1
Schaltet Textfenster ein.

WINDOW 0
Schaltet Textfenster aus.

Mit der Taste F5 wird der Cursor (der normale) in die obere linke Ecke des Textfensters gebracht. Dies entspricht der HOME-Taste für den gesamten Bildschirm. Die F5-Taste läßt sich ebenso programmieren wie die HOME-Taste, das heißt mit PRINT "(F5)" (auf dem Bildschirm erscheint ein reverses Grafikzeichen) läßt sich der Cursor in die obere linke Ecke des Textfensters bringen.

JOYSTICK ve
(ve=Verzögerung. Mit ve=0 malt man am schnellsten, mit ve=255 am langsamsten.)
Dieser Befehl erlaubt das Zeichnen von Bildern mit dem Joystick (Port 2). Mit dem Joystick kann der Grafikcursor bewegt werden. Druckt man gleichzeitig den Feuerknopf, wird entsprechend dem eingestellten Modus ein Punkt gesetzt, gelöscht etc. Nun können beliebige Bilder gezeichnet werden, so lange, bis mit der F7-Taste mit der Programmabarbeitung fortgefahren wird.

LPEN
Dieser Befehl erlaubt das Zeichnen von Bildern mit dem Lightpen (Port 1). Im Prinzip gilt das gleiche wie beim JOYSTICK-Befehl, nur daß hier statt des Feuerknopfes die CTRL-Taste beziehungsweise der Knopf am Lightpen benutzt wird. Man verläßt den Lightpenmodus mit F7.

Alle Befehle können unabhängig vom Einschaltzustand des Grafikbildschirms angewendet werden. Sie lassen sich auch wie normale Basic-Befehle abkürzen.

Die Funktionen und Anwendungen des Programms und seiner Befehle sind sehr gut in dem Programm »TURTLE DEMO« demonstriert (Listing 2).

Das Programm beginnt mit Titel und Autorenanschrift. Das GOSUB in Zeile 260 dient nur zur Suche von Syntax-Fehlern in den DATA-Zeilen. Im Programmteil »Variable« ab Zeilennummer 290 wird der Variablen AN die Startadresse des Maschinenprogramms zugewiesen, der Variable ZI die Endadresse. In NA$ steht der Programmname. Die eigentliche Einleseroutine (ab Zeile 350) funktioniert folgendermaßen: Es werden in einer Schleife die ersten 16 Zahlen aus einer jeden DATA-Zeile gelesen und in den Speicher gePOKEt. Gleichzeitig wird aus den gelesenen Daten eine Prüfsumme gebildet. Diese Summe wird mit der letzten Zahl in der DATA-Zeile verglichen, dies ist die richtige Prüfsumme. Unterscheiden sich die beiden Zahlen, so wurde ein Tippfehler gemacht und die fehlerhafte Zeile wird vom Programm automatisch geLISTet. Außerdem wird geprüft, ob eine Zeile vergessen wurde (Zeile 517-520) und ob die Anzahl der Daten richtig ist (Zeile 530-537). Zusätzlich wird noch getestet, ob die gelesene Zahl auch zwischen 0-255 liegt (Zeile 391). Ist dies nicht der Fall, wird eine Fehlermeldung ausgegeben (Zeile 503-505). Ursache ist wahrscheinlich ein Kommafehler. Wurde kein Fehler gefunden, so fragt das Programm, ob es sich selbst abspeichern soll (davon sollte bei der ersten Benutzung des Programms unbedingt Gebrauch gemacht werden, Zeile 546-580). Danach wird das Maschinenprogramm gestartet (Zeile 590-610). Trotz der Prüfsummen ist ein Fehler in den DATAs nicht völlig ausgeschlossen. Vertauschungen werden zum Beispiel nicht bemerkt. Solche Fehler sind jedoch sehr unwahrscheinlich.

Komfortabler DATA-Lader

Diese Art der Überprüfung von DATA-Werten sollten Sie sich genau ansehen. Sie erleichtert der Redaktion und vor allem den Lesern die Eingabe und Überprüfung großer Zahlenkolonnen. Falls Sie uns Programme einschicken wollen, dann nehmen Sie sich doch auch bitte etwas Zeit und fügen eine komfortable Prüfroutine in Ihre Programme ein.

Programmierung
a) Speicheraufteilung:
Das Video-RAM steht ab $0400. Der HiRes-Speicher wurde hinter $CC00 gelegt, das HiRes-Farb-RAM nach C800. Das Programm selbst beginnt bei $C000 und endet bei $C88B. Programmvariable liegen im Bereich vor $CC00.

b) Die Belegung der Funktionstasten wird durch ein »Anzapfen« des Interrupt ermöglicht. Die Funktion PRINT "(fs)" wird durch Verändern der BSOUT-Routine erreicht.

c) Das Textfenster:
Wie Sie wissen, wird das Bild auf dem Fernseher (Monitor) durch einen Elektronenstrahl erzeugt, der den Bildschirm zeilenweise von oben nach unten abfährt. Der VIC bietet nun die Möglichkeit bei einer bestimmten Zeilenposition einen Interrupt auszulösen. Diese Fähigkeit des VIC wird ausgenutzt. In der dann ausgeführten Interruptroutine wird zwischen HiRes- und Textmodus hin- und hergeschaltet, so daß der Bildschirm in ein Text- und ein Grafikfenster eingeteilt wird.

d) Die neuen Befehle:
Es gibt zahlreiche Möglichkeiten, neue Basic-Befehle zu implementieren. Anzapfen der

1. CHRGET-Routine (DOS 5.1)
2. Eingabe-Warteschleife (Toolkits)
3. Interpreterschleife (beschrieben im 64 Intern)

Die genannten Möglichkeiten haben jedoch alle große Nachteile: Sie sind alle entweder zu langsam (1. und 3.), die Befehle lassen sich nur im Direktmodus anwenden (2.) oder sie können nicht abgekürzt werden. Deshalb wurden die Befehle mit sehr komplexen Routinen voll als Tokens integriert. Aus diesem Grund wurde die Interpreterschleife, die Routine zur Umwandlung in Interpretercode und die Routine zur Rückumwandlung in Klartext verändert. Die Vorteile dieser Arbeitsweise sind, daß sich die neuen Befehle wie normale Basic-Befehle anwenden lassen und daß nur minimale Geschwindigkeitsverluste bei der Ausführung von Basic-Programmen in Kauf genommen werden müssen. Ähnliche Methoden verwenden übrigens auch professionelle Basic-Erweiterungen, wie zum Beispiel Simons Basic.

Einige Befehle definiert

Ein solches »Anzapfen« von Betriebssystem- und Basic-Routinen wird beim C64 durch zahlreiche Sprungvektoren ermöglicht, die verändert werden können. Dies ist leider nicht bei allen Computern so gut möglich. Die hier benutzten Vektoren sind:

Name:	Adresse in dezimal:
Interrupt-Vektor	: 788/789
NMI-Vektor	: 792/793
BRK-Vektor	: 790/791
Bsout-Vektor	: 806/807
Vektor für Umwandlung in Interpretercode	: 772/773
Vektor für Umwandlung in Klartext	: 774/775
Vektor für Basic-Befehl ausführen (Interpreterschleife)	: 776/777

Das Wissen über die Programmierung des Programms ist für dessen Anwendung nicht nötig. Doch nun viel Spaß mit Turtle-Grafik. (Peter Menke/ah)

Benutzte Variable:

AN	= Anfangsadresse des Maschinenprogramms
ZI	= Endadresse des Maschinenprogramms
NA$	= Programmname beim Abspeichern
GE	= Geräteadresse beim Abspeichern
X	= eingelesener DATA-Wert
S	= Speicherstelle, in die X gePOKEt wird
Z	= aktuelle DATA-Zeilennummer
PR	= errechnete Prüfsumme
I	= Variable der FOR/NEXT-Einleseschleife

```
20 PRINT"{CLR,2DOWN,4SPACE}BBBBBBBBBBBBBBB
   BBBBBBBBBBBBBBBB                     <031>
30 PRINT"{4SPACE}BBBBBBBBBBBBBBBBBBBBBBBBBBB
   BBBBBB                               <124>
40 PRINT"{4SPACE}BB{27SPACE}BB          <095>
50 PRINT"{4SPACE}BB{27SPACE}BB          <105>
60 PRINT"{4SPACE}BB{5SPACE}TURTLE GRAPHICS
   {7SPACE}BB                           <098>
70 PRINT"{4SPACE}BB{5SPACE}EEEEEEEEEEEEEEE
   {7SPACE}BB                           <213>
100 PRINT"{4SPACE}BB{27SPACE}BB         <155>
110 PRINT"{4SPACE}BB{3SPACE}ENTWICKELT 8.1
    984 VON{3SPACE}BB                   <090>
120 PRINT"{4SPACE}BB{27SPACE}BB         <175>
130 PRINT"{4SPACE}BB{27SPACE}BB         <185>
140 PRINT"{4SPACE}BB{3SPACE}P E T E R{2SPA
    CE}M E N K E{4SPACE}BB              <011>
150 PRINT"{4SPACE}BB{27SPACE}BB         <205>
160 PRINT"{4SPACE}BB{8SPACE}GRIFTWEG 4{9SP
    ACE}BB                              <202>
170 PRINT"{4SPACE}BB{4SPACE}2212 BRUNSBUET
    TEL{6SPACE}BB                       <120>
190 PRINT"{4SPACE}BB{27SPACE}BB         <245>
200 PRINT"{4SPACE}BB{27SPACE}BB         <255>
210 PRINT"{4SPACE}BBBBBBBBBBBBBBBBBBBBBBBBBB
    BBBBBBB                             <048>
220 PRINT"{4SPACE}BBBBBBBBBBBBBBBBBBBBBBBBBB
    BBBBBBB                             <058>
230 PRINT"{2DOWN,9SPACE}BITTE TASTE DRUECK
    EN"                                 <074>
240 GET A$:IF A$=""THEN 240             <146>
250 PRINT"{CLR}"                        <238>
260 GOSUB 1000                          <218>
270 :                                   <248>
280 :                                   <002>
290 REM **VARIABLE**                    <241>
300 AN=49152                            <213>
301 ZI=51360                            <046>
310 NA$="TURTLE GR."                    <122>
320 :                                   <042>
330 :                                   <052>
340 :                                   <062>
350 REM **** EINLESEROUTINE ****        <080>
360 S=AN:Z=1000                         <060>
370 FOR I=1 TO 16:READ X                <096>
390 IF X=-100 THEN 530                  <107>
391 IF X<0 OR X>255 THEN 503            <105>
400 GOSUB 515:POKE S,X                  <143>
410 S=S+1:PR=PR+X:NEXT                  <224>
440 :                                   <162>
460 READ X:IF PR=X THEN PR=0:Z=Z+10:GOTO 3
    70                                  <110>
470 PRINT"{2DOWN}PRUEFSUMMENFEHLER IN ZEIL
    E"Z                                 <155>
475 PRINT"{DOWN}PRUEFSUMME DER ZEILE"Z":"P
    R:PRINT                             <160>
480 PRINT"RICHTIGE PRUEFSUMME:"X:PRINT:PRI
    NT                                  <112>
490 PRINT"{DOWN}LIST"Z-10"-"Z"{2UP}";:POKE
     631,13:POKE 198,1                  <110>
500 END                                 <248>
501 :                                   <223>
502 :                                   <224>
503 PRINT"DATENFEHLER IN ZEILE"Z        <234>
504 PRINT"ES WURDE VERSUCHT"X"ZU POKEN."  <031>
505 GOTO 490                            <091>
510 :                                   <232>
511 :                                   <233>
512 :                                   <236>
515 PRINT"{HOME,2DOWN,2SPACE}ZEILE:"Z" SPE
    ICHERSTELLE:"S                      <219>
516 PRINT"{DOWN,2SPACE}EINGELESENER WERT:{
    4SPACE,4LEFT}"X                     <121>
517 IF Z>=PEEK(63)+256*PEEK(64) THEN RETUR
    N                                   <051>
518 PRINT"{3DOWN}DATA-ZEILE FEHLT!!"    <166>
519 PRINT"ODER ZEILENNUMMERN DER DATA-ZEIL
    EN NICHT"                           <191>
520 PRINT"IN ZEHNER ABSTAENDEN":END     <101>
521 :                                   <245>
522 :                                   <246>
523 :                                   <247>
530 IF S=ZI THEN 539                    <248>
531 PRINT"{3DOWN}DATENANZAHL IST FALSCH":P
    RINT:PRINT"DAS WAERE RICHTIG :";ZI-AN  <078>
532 PRINT:PRINT"IHRE DATENANZAHL :";S-AN  <083>
537 END                                 <031>
539 PRINT"{2DOWN}KEINEN FEHLER GEFUNDEN !{
    DOWN}"                              <239>
540 PRINT"ABSPEICHERN (J/N) ?"          <187>
550 GET A$:IF A$="N" THEN 590           <119>
560 IF A$<>"J" THEN 550                 <207>
570 INPUT"GERAETEADRESSE (1=DATASETTE/8=DI
    SK)";GE                             <233>
580 SAVE NA$+"(C)PM",GE                 <062>
590 PRINT"{DOWN}ZUM START EINE BELIEBIGE{1
    6SPACE}TASTE DRUECKEN               <064>
600 GET A$:IF A$="" THEN 600            <125>
610 SYS AN                              <206>
630 :                                   <098>
640 :                                   <108>
650 :                                   <118>
1000 DATA 120,032,063,193,088,032,163,193,
     032,223,193,169,028,133,254,169,2085  <212>
1010 DATA 192,133,255,032,097,192,032,068,
     166,076,116,164,147,017,017,017,1721  <190>
1020 DATA 029,029,084,085,082,084,076,069,
     032,071,082,065,080,072,073,067,1080  <078>
1030 DATA 083,190,190,029,029,066,089,032,
     080,069,084,069,082,032,077,069,1270  <017>
1040 DATA 078,075,069,190,190,029,029,051,
     056,057,049,049,032,066,065,083,1168  <168>
1050 DATA 073,067,032,066,089,084,069,083,
     032,070,082,069,069,017,017,190,1109  <134>
1060 DATA 191,160,000,177,254,201,191,240,
     012,201,190,208,002,169,013,032,2241  <230>
1070 DATA 210,255,200,208,238,096,173,025,
     208,141,025,208,016,029,173,018,2223  <050>
1080 DATA 208,201,250,176,011,032,204,193,
     169,250,141,018,208,076,188,254,2579  <238>
1090 DATA 032,185,193,169,217,141,018,208,
     076,188,254,173,013,220,088,165,2340  <009>
1100 DATA 204,208,033,169,032,044,017,208,
     240,026,206,252,203,208,021,169,2240  <231>
1110 DATA 007,141,252,203,173,241,203,072,
     169,002,141,241,203,032,118,195,2393  <017>
1120 DATA 104,141,241,203,165,212,208,030,
     165,216,208,026,165,203,205,255,2747  <224>
1130 DATA 203,240,019,141,255,203,201,064,
     240,012,201,004,240,011,201,005,2240  <185>
1140 DATA 240,029,201,006,240,048,076,049,
     234,173,017,208,041,032,240,009,1843  <177>
1150 DATA 032,204,193,032,017,194,076,230,
     192,032,185,193,076,230,192,173,2251  <254>
1160 DATA 017,208,041,032,240,246,173,026,
     208,041,001,240,006,032,017,194,1722  <007>
1170 DATA 076,230,192,032,254,193,162,021,
     160,000,024,032,240,255,076,230,2177  <056>
1180 DATA 192,072,138,072,152,072,169,127,
     141,013,221,172,013,221,048,012,1835  <193>
1190 DATA 032,163,253,032,024,229,032,063,
     193,108,002,160,076,114,254,032,1767  <149>
1200 DATA 204,193,169,004,141,136,002,169,
     118,141,020,003,169,192,141,021,1823  <018>
1210 DATA 003,169,033,141,024,003,169,193,
     141,025,003,169,048,141,022,003,1287  <029>
1220 DATA 169,193,141,023,003,169,011,141,
     254,203,169,001,141,253,203,141,2215  <123>
1230 DATA 252,203,169,000,141,026,208,169,
     134,141,004,003,169,196,141,005,1961  <029>
1240 DATA 003,169,043,141,008,003,169,197,
     141,009,003,169,084,141,006,003,1289  <090>
1250 DATA 169,197,141,007,003,169,103,141,
     038,003,169,199,141,039,003,032,1554  <170>
1260 DATA 249,195,096,162,032,160,000,169,
     224,133,255,132,254,152,145,254,2612  <155>
1270 DATA 136,208,251,230,255,202,208,246,
     096,169,056,141,024,208,169,148,2747  <203>
1280 DATA 141,000,221,173,017,208,009,032,
     141,017,208,096,169,021,141,024,1618  <170>
1290 DATA 208,169,151,141,000,221,173,017,
     208,041,223,141,017,208,096,169,2183  <170>
1300 DATA 204,133,255,160,000,132,254,162,
     004,173,253,203,010,010,010,010,1973  <232>
1310 DATA 013,254,203,145,254,136,208,251,
     230,255,202,208,246,096,169,217,3087  <129>
1320 DATA 141,018,208,173,017,208,041,127,
     141,017,208,169,001,141,026,208,1844  <006>
1330 DATA 096,169,000,141,026,208,096,166,
     122,202,232,142,242,203,162,255,2462  <221>
1340 DATA 142,251,203,142,250,203,238,250,
     203,174,250,203,189,127,194,240,3259  <094>
```

Listing 1. Turtle-Grafik.
Beachten Sie bitte die Eingabehinweise auf Seite 16.

```
1350 DATA 033,016,243,238,251,203,172,242,
     203,136,200,185,000,002,141,249,2514  <170>
1360 DATA 203,232,189,127,194,056,237,249,
     203,240,239,201,128,240,019,076,2833  <114>
1370 DATA 038,194,174,242,203,189,000,002,
     240,007,201,058,240,188,232,208,2416  <078>
1380 DATA 244,096,174,242,203,173,251,203,
     024,105,208,157,000,002,232,200,2514  <184>
1390 DATA 185,000,002,157,000,002,240,218,
     232,200,208,244,076,082,194,200,2240  <137>
1400 DATA 072,073,082,069,211,077,079,086,
     197,067,076,069,065,210,087,073,1593  <153>
1410 DATA 078,068,079,215,080,076,079,212,
     077,079,068,197,068,069,199,082,1726  <190>
1420 DATA 069,086,069,082,211,076,084,085,
     082,206,082,084,085,082,206,067,1656  <095>
1430 DATA 079,076,079,210,071,076,079,065,
     196,071,083,065,086,197,074,079,1586  <050>
1440 DATA 089,083,084,073,067,203,076,080,
     069,206,072,069,076,208,000,239,1694  <040>
1450 DATA 194,205,197,095,195,084,199,073,
     196,118,196,147,197,163,197,128,2584  <191>
1460 DATA 198,144,198,165,198,036,199,223,
     198,161,199,059,200,053,199,032,2462  <222>
1470 DATA 158,183,224,000,240,054,032,121,
     000,240,040,032,253,174,032,158,1941  <180>
1480 DATA 183,224,016,144,003,076,106,195,
     142,254,203,032,223,193,032,121,2147  <124>
1490 DATA 000,240,016,032,253,174,032,158,
     183,224,016,144,003,076,106,195,1852  <221>
1500 DATA 142,032,208,032,185,193,032,249,
     195,076,174,167,032,121,000,240,2078  <153>
1510 DATA 037,032,253,174,032,158,183,224,
     016,144,003,076,106,195,142,033,1808  <159>
1520 DATA 208,032,121,000,240,016,032,253,
     174,032,158,183,224,016,144,003,1836  <025>
1530 DATA 076,106,195,142,032,208,032,017,
     194,032,204,193,076,174,167,208,2056  <019>
1540 DATA 012,032,163,193,032,223,193,076,
     174,167,162,014,044,162,011,032,1690  <149>
1550 DATA 204,193,076,055,164,096,173,241,
     203,201,003,240,248,174,246,203,2720  <239>
1560 DATA 138,074,074,041,254,168,185,015,
     196,141,244,203,185,016,196,141,2271  <047>
1570 DATA 245,203,138,041,007,024,109,244,
     203,141,244,203,173,247,203,041,2466  <079>
1580 DATA 248,141,243,203,173,244,203,133,
     252,024,169,224,109,245,203,133,2947  <000>
1590 DATA 253,024,165,252,109,243,203,133,
     252,165,253,109,248,203,133,253,2998  <152>
1600 DATA 173,247,203,041,007,073,007,170,
     189,065,196,160,000,120,162,052,1865  <096>
1610 DATA 134,001,174,241,203,240,008,224,
     001,240,014,224,002,240,019,017,1982  <250>
1620 DATA 252,145,252,169,055,133,001,088,
     096,073,255,049,252,145,252,076,2293  <232>
1630 DATA 227,195,081,252,145,252,076,227,
     195,169,000,141,248,203,141,241,2793  <116>
1640 DATA 203,141,240,203,169,160,141,247,
     203,169,100,141,246,203,096,000,2662  <004>
1650 DATA 000,064,001,128,002,192,003,000,
     005,064,006,128,007,192,008,000,800   <025>
1660 DATA 010,064,011,128,012,192,013,000,
     015,064,016,128,017,192,018,000,880   <035>
1670 DATA 020,064,021,128,022,192,023,000,
     025,064,026,128,027,192,028,000,960   <149>
1680 DATA 030,001,002,004,008,016,032,064,
     128,032,235,183,224,200,176,035,1370  <090>
1690 DATA 165,021,201,001,144,008,208,027,
     165,020,201,064,176,021,120,142,1684  <135>
1700 DATA 246,203,165,020,141,247,203,165,
     021,141,248,203,088,032,118,195,2436  <219>
1710 DATA 076,174,167,076,106,195,032,158,
     183,224,004,144,003,076,106,195,1919  <007>
1720 DATA 142,241,203,076,174,167,032,023,
     194,166,122,160,004,132,015,189,2040  <095>
1730 DATA 000,002,141,249,203,041,240,201,
     208,208,006,173,249,203,076,232,2432  <117>
1740 DATA 196,173,249,203,016,007,201,255,
     240,062,232,208,226,201,032,240,2741  <242>
1750 DATA 055,133,008,201,034,240,086,036,
     015,112,045,201,063,208,004,169,1610  <248>
1760 DATA 153,208,037,201,048,144,004,201,
     060,144,029,132,113,160,000,132,1766  <022>
1770 DATA 011,136,134,122,202,200,232,189,
     000,002,056,249,158,160,240,245,2336  <244>
1780 DATA 201,128,208,048,005,011,164,113,
     232,200,153,251,001,185,251,001,2152  <186>
1790 DATA 240,054,056,233,058,240,004,201,
     073,208,002,133,015,056,233,085,1891  <108>
1800 DATA 208,141,133,008,189,000,002,240,
     223,197,008,240,219,200,153,251,2412  <110>
1810 DATA 001,232,208,240,166,122,230,011,
     200,185,157,160,016,250,185,158,2521  <046>
1820 DATA 160,208,180,189,000,002,016,190,
     076,009,166,032,115,000,041,240,1624  <101>
1830 DATA 201,208,240,009,032,121,000,032,
     237,167,076,174,167,032,121,000,1817  <081>
1840 DATA 041,015,010,168,185,208,194,133,
     167,185,207,194,133,166,032,115,2153  <053>
1850 DATA 000,108,166,000,016,052,201,255,
     240,048,036,015,048,044,141,249,1619  <026>
1860 DATA 203,041,240,201,208,208,038,173,
     249,203,041,015,170,232,132,073,2427  <064>
1870 DATA 160,255,200,185,127,194,016,250,
     202,208,247,200,185,127,194,048,2798  <010>
1880 DATA 006,032,071,171,076,123,197,076,
     239,166,076,243,166,173,249,203,2267  <161>
1890 DATA 076,036,167,032,158,183,224,008,
     144,003,076,106,195,142,240,203,1993  <203>
1900 DATA 076,174,167,208,037,162,032,169,
     224,133,255,160,000,132,254,120,2303  <232>
1910 DATA 169,052,133,001,177,254,073,255,
     145,254,136,208,247,230,255,202,2791  <027>
1920 DATA 208,242,169,055,133,001,088,076,
     174,167,076,109,195,032,138,173,2036  <190>
1930 DATA 032,247,183,166,020,240,008,032,
     236,197,198,020,076,211,197,166,2229  <240>
1940 DATA 021,240,006,198,021,198,020,208,
     234,076,174,167,173,240,203,240,2419  <185>
1950 DATA 029,201,002,240,034,201,004,240,
     039,201,006,240,044,201,001,240,1923  <018>
1960 DATA 010,201,003,240,015,201,005,240,
     020,208,027,032,047,198,032,093,1572  <186>
1970 DATA 198,076,118,195,032,067,198,032,
     047,198,076,118,195,032,056,198,1836  <019>
1980 DATA 032,067,198,076,118,195,032,093,
     198,032,056,198,076,118,195,173,1857  <233>
1990 DATA 246,203,240,067,206,246,203,096,
     173,246,203,201,199,176,056,238,2999  <144>
2000 DATA 246,203,096,173,248,203,208,009,
     173,247,203,240,042,206,247,203,2947  <094>
2010 DATA 096,173,247,203,208,003,206,248,
     203,206,247,203,096,173,248,203,2963  <013>
2020 DATA 208,009,238,247,203,208,003,238,
     248,203,096,174,247,203,232,224,2981  <068>
2030 DATA 064,176,004,142,247,203,096,173,
     237,203,208,003,076,106,195,096,2229  <144>
2040 DATA 032,158,183,138,024,109,240,203,
     041,007,141,240,203,076,174,167,2136  <006>
2050 DATA 032,158,183,142,249,203,173,240,
     203,056,237,249,203,041,007,141,2517  <189>
2060 DATA 240,203,076,174,167,032,158,183,
     224,016,176,048,142,253,203,032,2327  <201>
2070 DATA 223,193,032,121,000,240,034,032,
     253,174,032,158,183,224,016,176,2091  <028>
2080 DATA 027,142,254,203,032,223,193,032,
     121,000,240,013,032,253,174,032,1971  <090>
2090 DATA 158,183,224,016,176,006,142,032,
     208,076,174,167,076,106,195,120,2059  <213>
2100 DATA 169,052,133,001,162,032,160,000,
     132,254,132,166,169,224,133,255,2174  <096>
2110 DATA 169,160,133,167,177,254,145,166,
     136,208,249,230,255,230,167,202,3048  <222>
2120 DATA 208,242,169,055,133,001,088,032,
     212,225,162,000,160,192,134,254,2267  <185>
2130 DATA 169,160,133,255,169,054,133,001,
     169,254,032,216,255,169,055,133,2357  <028>
2140 DATA 001,076,174,167,032,212,225,169,
     000,133,185,162,000,160,224,032,1952  <069>
2150 DATA 213,255,076,174,167,208,026,160,
     000,185,128,194,240,016,016,007,2065  <040>
2160 DATA 041,127,032,210,255,169,013,032,
     210,255,200,076,057,199,076,174,2126  <187>
2170 DATA 167,076,109,195,032,158,183,224,
     000,240,006,032,254,193,076,174,2119  <033>
2180 DATA 167,032,017,194,076,174,167,072,
     165,154,201,003,208,008,104,201,1943  <209>
2190 DATA 135,240,006,076,022,231,076,213,
     241,142,249,203,166,212,208,027,2447  <072>
2200 DATA 166,216,208,023,152,072,169,021,
     133,214,169,000,133,211,032,108,2027  <011>
2210 DATA 229,104,168,174,249,203,169,135,
```

Listing 1. Turtle-Grafik (Fortsetzung)

```
     024,088,096,174,249,203,076,022,2363  <034>
2220 DATA 231,032,158,183,232,142,239,203,
     169,001,141,237,203,165,203,201,2740  <254>
2230 DATA 003,208,008,169,000,141,237,203,
     076,174,167,169,000,141,238,203,2137  <055>
2240 DATA 173,000,220,041,001,208,009,032,
     047,198,032,034,200,238,238,203,1874  <076>
2250 DATA 173,000,220,041,002,208,009,032,
     056,198,032,034,200,238,238,203,1884  <135>
2260 DATA 173,000,220,041,004,208,009,032,
     067,198,032,034,200,238,238,203,1897  <225>
2270 DATA 173,000,220,041,008,208,009,032,
     093,198,032,034,200,238,238,203,1927  <075>
2280 DATA 173,238,203,208,168,032,011,200,
     076,173,199,173,241,203,072,169,2539  <214>
2290 DATA 002,141,241,203,032,118,195,032,
     047,200,032,118,195,104,141,241,2042  <236>
2300 DATA 203,096,173,000,220,041,016,208,
     226,032,118,195,076,047,200,172,2023  <028>
2310 DATA 239,203,162,100,202,208,253,136,
     208,248,096,240,003,076,109,195,2678  <096>
2320 DATA 165,203,201,003,208,003,076,174,
     167,173,020,208,056,233,050,144,2084  <087>
2330 DATA 239,201,200,176,235,141,246,203,
     173,019,208,056,233,035,144,224,2733  <233>
2340 DATA 010,141,247,203,144,007,201,064,
     176,214,169,001,044,169,000,141,1931  <006>
2350 DATA 248,203,173,141,002,041,004,240,
     012,032,118,195,032,093,198,032,1764  <040>
2360 DATA 118,195,076,064,200,173,241,203,
     072,169,002,141,241,203,032,118,2248  <140>
2370 DATA 195,032,118,195,104,141,241,203,
     076,064,200,255,255,255,255,249,2838  <071>
63000 DATA -100:RETURN                     <076>
```

© 64'er

Listing 1. Turtle-Grafik (Schluß)

Listing 2. Demo-Programm zur Turtle-Grafik

```
0 REM T U R T L E   D E M O
1 REM BY PETER MENKE
2 :
3 HIRES 1,0,0:COLOR15:CLEAR
4 :
10 REM ROBOTER-KOPF
20 PLOT 120,160
30 DEG 2:MOVE 10
40 DEG 0:MOVE 30
50 DEG 2:MOVE 10
55 DEG 4:MOVE 40
60 DEG 2:MOVE 100
65 DEG 0:MOVE 100
70 DEG 6:MOVE 100
90 DEG 4:MOVE 40
100 DEG 6:MOVE 10
110 DEG 0:MOVE 30
120 DEG 6:MOVE 10
121 :
122 REM LINKES OHR
130 PLOT 110,105
140 DEG 4:MOVE 15
150 DEG 2:MOVE 30
160 DEG 0:MOVE 15
161 :
162 REM RECHTES OHR
170 PLOT 210,105
180 DEG 0:MOVE 15
190 DEG 2:MOVE 30
200 DEG 4:MOVE 15
201 :
202 REM MUND
210 PLOT 130,120
220 DEG 6:MOVE 10
230 DEG 0:MOVE 60
240 DEG 2:MOVE 10
241 :
242 REM NASE
250 PLOT 152,100
260 DEG 0:MOVE 16
270 DEG 3:MOVE 8
280 DEG 4:MOVE 1
290 DEG 5:MOVE 8
291 :
292 REM LINKES AUGE
293 PLOT 130,60:DEG 0
295 FORT=1TO4
300 MOVE 15
310 RTURN2
320 NEXT
330 :
340 REM RECHTES AUGE
350 PLOT 190,60:DEG 6
360 FORT=1TO4
370 MOVE 15
380 RTURN 2
390 NEXT
400 :
410 WINDOW 1:PRINT"⊐▌                T U R T L
E   D E M O"
420 PRINT"▩               BY PETER MENKE"
430 FORI=0TO4000:NEXT
440 WINDOW 0
450 :
460 :
461 REM SPIRALE
470 HIRES 1,2,2:COLOR 0:CLEAR:PLOT 160,1
00
480 FORI=1TO66
490 LTURN 1
500 MOVE I
510 NEXT
511 FORI=1TO1000:NEXT
520 :
530 :
540 REM VIERECK-SPIRALEN
541 CLEAR:COLOR1
550 HIRES 1,2,2:FORI=1TO200STEP2
560 RTURN 2:MOVE I:NEXT
561 PLOT 160,100
570 FORI=1TO195STEP2
580 LTURN 2:MOVE I:NEXT
590 PLOT 160,100
600 FORI=1TO195
610 RTURN 2:MOVE I:NEXT
620 :
630 :
640 REM MUSTER
641 HIRES1,5,5:CLEAR:COLOR0
650 FORI=1TO45
660 FORX=0TO7
670 RTURNX
671 MOVE I
680 NEXTX
700 NEXT
710 :
720 HIRES1,5,5
730 FORI=1TO45
740 FORX=0TO7
750 LTURNX
761 MOVEI
780 NEXTX
790 NEXT
800 FORI=1TO2000:NEXT
810 :
820 :
```

Listing 2. Demo-Programm zur Turtle-Grafik (Fortsetzung)

```
830 REM PYRAMIDEN
840 HIRES1,6,6:CLEAR:COLOR15
850 FORI=1TO33
860 FORX=0TO7
870 DEG X
871 MOVE I
880 NEXTX
900 NEXT
910 :
930 FORI=1TO33
940 FORX=0TO7
950 DEG 7-X
961 MOVE I
980 NEXTX
990 NEXT
991 FORI=1TO8:REVERS
992 FORX=1TO500:NEXT
993 NEXT
1000 :
1010 :
1011 REM INVERSES MUSTER
1020 HIRES7,7:COLOR 0:CLEAR
1030 FORY=1TO4:HIRES 1:MODE 2
1040 FORI=1TO100
1050 MOVE I
1060 FORX=1TO7
1070 LTURN X:MOVEX:NEXTX,I,Y
1080 :
1090 :
1111 REM STRICH-MUSTER
1120 HIRES1,8,8:COLOR 0:CLEAR
1130 FORY=1TO4:HIRES 1:MODE2
1140 FORI=1TO45
1150 MOVE I
1160 FORX=1TO10
1170 RTURN X:MOVE X:NEXTX,I,Y
1181 :
1182 :
1183 REM JOYSTICK ZEICHNEN
1184 HIRES 1,10,10:COLOR0:CLEAR
1185 WINDOW1
1186 PRINT"⌐▮     JOYSTICK-ZEICHNEN (PORT
 2)"
1187 PRINT"     WEITER MIT F7"
1188 JOYSTICK5
1189 :
1190 :
1191 REM LIGHTPEN ZEICHNEN
1192 HIRES 1,1,1:COLOR0:CLEAR
1193 WINDOW1
1194 PRINT"⌐▮     LIGHTPEN-ZEICHNEN (PORT
 1)"
1195 PRINT"     WEITER MIT F7"
1196 LPEN
1197 :
1198 :
1200 REM ZUFALL
1210 HIRES 1:CLEAR:COLOR 1,9,9
1220 WINDOW 1
1230 PRINT"⌐▮ZUFALLSMUSTER
1240 PRINT"SCHLUSS MIT RUN/STOP-TASTE
1250 DEG RND(1)*8:MOVE 1:GOTO1250

READY.
```

Listing 2. Demo-Programm zur Turtle-Grafik (Schluß)

HI-EDDI Druckerroutinen

Jetzt kommen auch Besitzer von Commodore-Druckern in den Genuß, HI-EDDI-Bilder ausgeben zu können.

Welcher 64'er-Leser kennt ihn nicht — den HI-EDDI, das fantastische Zeichen- und Malprogramm. Jeder der einen Epson oder dazu kompatiblen Drucker besitzt, kommt sogar in den Genuß der ausgezeichneten Hardcopy-Funktion. Jetzt folgen Anpassungen für Drucker anderer Hersteller, für die Commodore-Drucker MPS 801/802/803 und kompatible.

HI-EDDI für MPS 802 und kompatible

Ich begann nach Überwindung meiner Abneigung gegen stundenlanges Tastenklopfen HI-EDDI abzutippen. Nach den ersten Tests dieses fantastischen Zeichen- und Malprogramms, bereute ich es aber nicht mehr mich dieser Tortur unterzogen zu haben. Doch mußte ich als Besitzer eines MPS-802-Druckers feststellen, was ist HI-EDDI ohne Superhardcopy?

Die Antwort darauf veranlaßte mich dazu, die Anregung in der Funktionsbeschreibung aufzugreifen und eine Lösung für einen anderen Drucker zu suchen. Gesagt — getan, hier ist mein Lösungsvorschlag für den Commodore MPS-802 (Listing 1).

Die Druckerroutine hat dieselben Eigenschaften wie die von Hans Haberl geschriebene Routine für Epson-Drucker. Daher kann auf eine Funktionsbeschreibung verzichtet werden. Ich hoffe, daß der Ausdruck von HI-EDDI-Grafikbildern mit diesem Programm auf allen Versionen des Commodore 1526/MPS-802 Druckers funktioniert. Bei meiner Ausführung (REV 07C) konnte ich bis jetzt keine Fehler feststellen.

(Franz Illetschko/ah)

HI-EDDI für MPS 801/803 und kompatible

Eine Lösung für den Commodore MPS 801/803 schickte uns Florian Kulzer (Listing 2). Auch hier braucht am HI-EDDI selbst nichts verändert zu werden. Die Hardcopy wird mit den Tasten »Commodore« und »P« gestartet.

Allerdings ist mit diesem Programm das Nebeneinanderdrucken zweier Bilder nicht möglich. Ausgegeben wird immer eine Hardcopy des aktuellen Bildschirms. Daher können die Fragen nach Nummer und Format der Hardcopy beliebig beantwortet werden. (Florian Kulzer/ah)

```
programm : hi-print       0d00 0f9c
------------------------------------

0d00 : 85 5b a9 64 a2 04 a0 00  f7
0d08 : 20 08 0e a9 6a a2 04 a0  f2
0d10 : 06 20 08 0e a2 6a 20 c9  7c
0d18 : ff a9 14 20 d2 ff a2 64  75
0d20 : 20 c9 ff a9 8d 20 d2 ff  7f
0d28 : a9 69 a2 04 a0 05 20 08  72
0d30 : 0e a5 5b 0a 90 03 4c 86  88
0d38 : 0e a5 5b 29 07 a2 02 20  e3
0d40 : 13 0e a9 14 85 fb a5 5b  cd
0d48 : 29 c0 f0 10 a5 5b 4a 4a  02
0d50 : 4a 29 07 a2 00 20 13 0e  ae
0d58 : a9 00 85 fb a9 00 8d 3d  2d
0d60 : 03 20 2d 0e 20 66 0e f0  d0
0d68 : 29 a2 69 20 c9 ff a2 00  68
0d70 : bd 3e 03 20 d2 ff e8 e0  a4
0d78 : 08 d0 f5 a9 0d 20 d2 ff  b8
0d80 : a2 64 20 c9 ff 20 74 0e  84
0d88 : a9 fe 20 d2 ff a9 8d 20  d6
0d90 : d2 ff 18 a5 5e 69 08 85  79
0d98 : 5e 90 02 e6 5f 20 e1 ff  1a
0da0 : f0 3f ee 3d 03 ad 3d 03  2c
0da8 : c9 28 d0 b5 a5 5b 29 c0  cb
0db0 : f0 18 a5 5e a6 5c 85 5c  fe
0db8 : 86 5e a5 5f a6 5d 85 5d  e9
0dc0 : 86 5f a5 fb 49 28 85 fb  c2
0dc8 : d0 92 a2 64 20 c9 ff a9  ba
0dd0 : 0d 20 d2 ff ee 3c 03 ad  da
0dd8 : 3c 03 c9 19 f0 03 4c 5c  3c
0de0 : 0d a9 69 20 c3 ff a2 6a  bc
0de8 : 20 c9 ff a9 24 20 d2 ff  b0
0df0 : a2 64 20 c9 ff a9 8d 20  c9
0df8 : d2 ff a9 6a 20 c3 ff 20  e2
0e00 : cc ff a9 64 20 c3 ff 60  a4
0e08 : 20 ba ff a9 00 20 bd ff  b2
0e10 : 4c c0 ff a0 06 d9 96 1f  98
0e18 : f0 03 88 d0 f8 a9 00 95  ce
0e20 : 5c 8d 3c 03 18 69 20 88  11
0e28 : 10 fb 95 5d 60 78 a9 34  20
0e30 : 85 01 a2 00 8a 9d 3e 03  73
0e38 : e8 e0 08 d0 f8 a9 80 85  96
0e40 : 60 a0 00 b1 5e a2 00 0a  36
0e48 : 90 0a 48 bd 3e 03 05 60  78
0e50 : 9d 3e 03 68 e8 e0 08 d0  32
0e58 : ee 46 60 c8 c0 08 d0 e3  f2
0e60 : a9 37 85 01 58 60 a2 00  39
0e68 : 8a dd 3e 03 d0 05 e8 e0  6b
0e70 : 08 d0 f6 60 18 a5 fb 6d  23
0e78 : 3d 03 aa f0 08 a9 20 20  8e
0e80 : d2 ff ca d0 fa 60 a5 5b  1e
0e88 : 29 07 a2 00 20 13 0e 85  bb
0e90 : 5f a9 00 85 5e 85 fb 85  82
0e98 : 59 85 5a a9 c0 95 fc 4a  c1
0ea0 : 4a e8 e0 04 d0 f7 a9 00  8a
0ea8 : 8d 3d 03 20 2d 0e a5 5a  27
0eb0 : f0 11 a2 00 bd 3e 03 0a  bf
0eb8 : 0a 0a 0a 9d 3e 03 e8 e0  5f
0ec0 : 08 d0 f1 20 62 0f 4c d6  2e
0ec8 : 0e a2 00 bd 4a 03 9d 46  9f
0ed0 : 03 e8 e0 04 d0 f5 20 8d  58
0ed8 : 0f f0 2f a2 69 20 c9 ff  3e
0ee0 : a2 00 bd 46 03 20 d2 ff  37
0ee8 : bd 46 03 20 d2 ff e8 e0  20
0ef0 : 04 d0 ef a9 0d 20 d2 ff  aa
0ef8 : a2 64 20 c9 ff 20 74 0e  fc
0f00 : a9 fe 20 d2 ff a9 8d 20  4e
0f08 : d2 ff 20 e1 ff d0 03 4c  49
0f10 : e1 0d ee 3d 03 a5 59 49  30
0f18 : 08 85 59 d0 ac 18 a5 5e  32
0f20 : 69 08 85 5e 90 02 e6 5f  2e
0f28 : ad 3d 03 c9 50 f0 03 4c  9f
0f30 : ab 0e a2 64 20 c9 ff a9  bb
0f38 : 0d 20 d2 ff a5 5a 49 19  8e
0f40 : 85 5a a5 5e a6 5c 85 5c  44
0f48 : 86 5e a5 5f a6 5d 85 5d  79
0f50 : 86 5f ee 3c 03 ad 3c 03  5e
0f58 : c9 32 f0 03 4c a6 0e 4c  a2
0f60 : e1 0d a9 00 aa a8 9d 46  25
0f68 : 03 e8 e0 08 d0 f8 aa b9  0b
0f70 : 3e 03 0a 90 0b 48 18 b9  8b
0f78 : 46 03 75 fc 99 46 03 68  e5
0f80 : e8 e0 04 d0 ed a2 00 c8  79
0f88 : c0 08 d0 e3 60 a2 00 8a  2d
0f90 : dd 46 03 d0 05 e8 e0 04  8e
0f98 : d0 f6 60 00 ff ff ff ff  fb
```

Listing 1. HI-EDDI und MPS 802.
Bitte beachten Sie die Eingabehinweise auf Seite 18.

```
programm : hi-print       0d00 0e22
------------------------------------

0d00 : a9 7f a2 04 a0 00 20 ba  92
0d08 : ff 20 c0 ff a2 7f 86 67  56
0d10 : 20 c9 ff a9 ff 85 61 a9  4f
0d18 : 07 85 fd a9 1c 85 97 a9  36
0d20 : 00 8d 12 0e a9 28 8d 14  67
0d28 : 0e a2 04 bd b6 0d 20 d2  3a
0d30 : ff ca 10 f7 a9 00 85 63  0f
0d38 : 85 64 ad 12 0e 85 65 a9  93
0d40 : 00 85 fe a5 63 a6 64 a4  bd
0d48 : 65 20 bb 0d a0 00 b1 ac  78
0d50 : a6 fe 9d 16 0e e6 65 e8  1f
0d58 : 86 fe e4 fd d0 e5 a9 00  39
0d60 : a0 07 a6 fd 1e 16 0e 2a  0c
0d68 : ca 10 f9 25 61 09 80 20  fe
0d70 : d2 ff 88 10 ed a5 63 18  30
0d78 : 69 08 85 63 90 02 e6 64  31
0d80 : ce 14 0e d0 b5 a9 0d 20  13
0d88 : d2 ff ad 12 0e 18 69 07  5d
0d90 : 8d 12 0e c6 97 f0 03 4c  28
0d98 : 24 0d a9 04 c5 fd f0 0c  56
0da0 : 85 fd a9 01 85 97 a9 0f  88
0da8 : 85 61 d0 eb a9 0f 20 d2  c9
0db0 : ff a9 7f 4c cc ff 50 00  fb
0db8 : 10 1b 08 85 14 86 15 98  04
0dc0 : 4a 4a 4a aa bd f4 0d 85  da
0dc8 : ad 8a 29 03 aa bd 0e 0e  52
0dd0 : 85 ac 98 29 07 18 65 ac  17
0dd8 : 85 ac a5 14 29 f8 85 63  d6
0de0 : a9 20 05 ad 85 ad 18 a5  02
0de8 : ac 65 63 85 ac a5 ad 65  4a
0df0 : 15 85 ad 60 00 01 02 03  55
0df8 : 05 06 07 08 0a 0b 0c 0d  07
0e00 : 0f 10 11 12 14 15 16 17  0e
0e08 : 19 1a 1b 1c 1e 1f 00 40  d4
0e10 : 80 c0 a1 00 11 00 00 00  6a
0e18 : 00 80 00 00 00 00 00 ff  58
0e20 : 00 ff 02 c8 cc 00 02 90  af
```

Listing 2. HI-EDDI und MPS 801.
Bitte beachten Sie die Eingabehinweise auf Seite 18.

Hi-EDDI auf Star SG-10

Betrifft: Anpassung der Original-Druckerroutine HI-PRINT von HI-EDDI an den SG-10 von Star.

Folgende Änderungen müssen an der Druckerroutine vorgenommen werden:

460 DATA 4: REM Direktkanal

550 DATA 27,51,15,255,255: REM Zeilenabstand 15/144 Inch

580 DATA 27,103,4,255,255: REM Grafikmodus 4

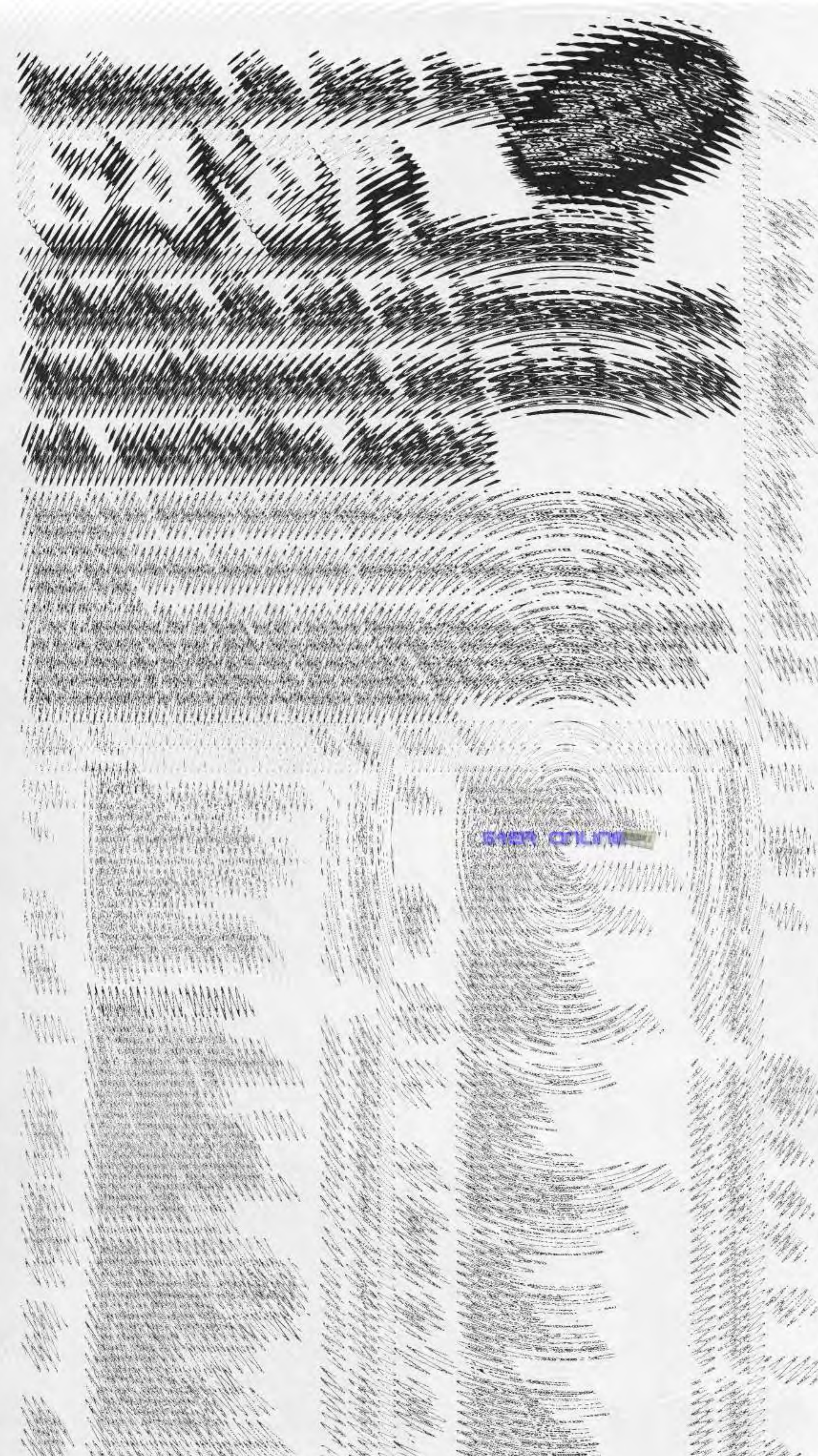
In dieser Betriebsart werden Kreise in Y-Richtung etwas gedehnt. Wer auf den Ausdruck zweier Bilder nebeneinander und auf Großbilder verzichtet, sollte folgende Werte eingeben (Zeile 460 muß natürlich trotzdem geändert werden):

550 DATA 27,51,16,255,255: REM Zeilenabstand 16/144 Inch

580 DATA 27,103,5,255,255: REM Grafikmodus 5

Dafür werden Kreise auch als Kreise dargestellt. Wenn man breiteres Papier als A4 verwendet, kann man auch zwei Bilder nebeneinander und Großbilder drucken. Der Drucker muß auf Star-Betriebsart eingestellt sein (siehe Handbuch). Der User-Port kann weiterhin als Parallelschnittstelle verwendet werden.

(Kurt Reiter/tr)





HI-EDDI und Simons Basic

Kleine Programme ermöglichen es, mit Simons Basic erstellte Bilder in HI-EDDI zu laden.

Simons Basic bietet leider keine Möglichkeit, High-Resolution-Bilder auf Diskette zu speichern und zu laden. Dies ist jedoch Voraussetzung, um Bilder, die mit Simons Basic erstellt wurden, auch in HI-EDDI und andere Programme zu laden.

Ich habe deshalb das Programm »Screensave« (Listing1) geschrieben, das die Simons-Basic-Befehle »SCRSV« und »SCRLD« um die Fähigkeit, Hires-Bilder zu laden und zu speichern, erweitert. Leider funktioniert das nicht für die Modul-Version, da direkt das Simons Basic modifiziert wird. Dafür wird kein Speicherplatz außerhalb des Simons Basic gebraucht. Außerdem ist die modifizierte Version abspeicherbar, nach ihrem Laden stehen die erweiterten Befehle sofort zur Verfügung, ohne ein weiteres Programm nachladen zu müssen.

Hinweise zum Eintippen:

- Simons Basic starten
- »Screensave« eintippen, sicherheitshalber abspeichern und starten
- Nachdem alle DATA-Fehler beseitigt wurden, stehen die erweiterten Befehle »SCRSV« und »SCRLD« zur Verfügung: Die Syntax ist dieselbe wie bisher, mit der Sekundäradresse wird jetzt der Bildschirm gewählt:

Sekundäradresse = 2: Low-Resolution-Bildschirm,
Sekundäradresse = 3: High-Resolution-Bildschirm

Das Aufzeichnungsformat des Low-Resolution-Bildschirmes ist leider nicht kompatibel mit dem der alten Befehle. Das des High-Resolution-Bildschirmes ist dasselbe wie bei HI-EDDI, Diashow und anderen Grafikprogrammen. Dazu muß das Bild aber unbedingt als PGM-File, also zum Beispiel mit:
SCRSV1,8,3,"NAME,P,W"

```
1 REM******        SCREENSAVE        ******      <192>
10 DATA 32,3,178,166,184,32,201,255,152,32
   ,210,255,165,252,32,210,255,169              <217>
20 DATA 210,133,248,162,32,36,250,48,2,162
   ,4,36,250,48,12,177,253,32,0,178             <165>
30 DATA 145,253,177,251,56,176,14,120,169,
   52,133,1,177,251,72,169,54,133               <104>
40 DATA 1,88,104,32,0,178,145,251,200,208,
   29,230,252,230,254,202,208,22,240            <075>
50 DATA 26,234,32,3,178,166,184,32,198,255
   ,32,207,255,32,207,255,169,207               <176>
60 DATA 208,180,234,169,195,36,144,240,183
   ,32,204,255,165,184,32,195,255               <077>
70 DATA 76,42,130,108,248,0,32,219,131,32,
   35,129,160,0,132,251,132,253,165             <107>
80 DATA 185,106,106,133,250,48,8,169,216,1
   33,254,169,4,208,2,169,224,133               <220>
90 DATA 252,169,255,133,249,96,234              <215>
100 S=0:FOR I=45455 TO 45608:READ A:S=S+A:
    POKE I,A:NEXT I                             <253>
110 IF S<>22096 THEN PRINT"FEHLER IN DATA
    S":END                                      <061>
120 PRINT"OK"                                   <212>
© 64'er
```

Listing 1. Screensave: Erweiterung für Simons Basic

abgespeichert werden, da HI-EDDI keine SEQ-Files laden kann.

Zum Abspeichern des »neuen« Simons Basic dient das Programm »Saver« (Listing 2). Dieses Programm speichert nicht nur das komplette Simons Basic, das übrigens auch noch andere Modifikationen, zum Beispiel die aus 64'er Ausgabe 11/84, enthalten kann, sondern auch die Funktionstastenbelegung mit ab. Nach dem Laden der mit »Saver« erstellten Simons-Basic-Version stehen somit nicht nur die neuen Befehle, sondern auch die individuelle Funktionstastenbelegung sofort zur Verfügung.

Hinweise zum Eintippen:

- Funktionstasten nach eigenen Wünschen belegen
- »Saver« eintippen, abspeichern, eine Diskette mit mindestens 66 freien Blocks einlegen und Programm starten. Nun fragt der »Saver« nach dem Namen, unter dem er die neue Version ablegen soll. Diese neue Version ist um einen Block länger als die alte (66 statt 65), da die Funktionstastenbelegung hinzugekommen ist.

Tips zu HI-EDDI

Die Anleitung in den REM-Zeilen des Programms»HI-PRINT.DATA« ist zugegebenermaßen etwas dürftig ausgefallen, so daß eine ausführlichere Erklärung dringend nötig ist:

Wie schon in der Bedienungsanleitung erwähnt, beschränkt sich die Anpaßbarkeit dieses Programms auf 8-Nadel-Drucker mit »vernünftiger« Einzelnadelansteuerung, wie zum Beispiel Drucker von Epson oder Mannesmann (im Gegensatz zum 1526/MPS 802). Mit dem Datum in Zeile 420 wird zwischen dem seriellen Bus (also Hardware-Interface) oder dem User-Port gewählt. Im ersten Fall muß mit den Daten in Zeile 440 (Primäradresse) und 460 (Sekundäradresse) das Interface auf Direktmodus gestellt werden. Hardwareinterfaces bieten in der Regel verschiedene Betriebsmodi (zum Beispiel Emulation eines Commodore-Druckers, spezielle List- oder Grafikmodi und eben den Direktmodus oder Linearmodus), die meist über die Sekundäradresse angewählt werden. »HI-PRINT« braucht den Direktmodus, in dem sich das Interface wie ein einfaches Kabel verhält und die Byte vom Computer unverändert an den Drucker übergibt. Für das Data-Becker-Interface muß zum Beispiel die Sekundäradresse 1 eingegeben werden, für das Görlitz-Interface 4 oder 12 (ohne beziehungsweise mit Auto-Linefeed). Beim Anschluß an den User-Port sind die Daten in Zeile 440 und 460 zwar nicht relevant, man darf sie jedoch nicht weglassen, da sonst die nachfolgenden DATAs verschoben würden.

Aus dem gleichen Grund dürfen auch die Längen der folgenden DATA-Zeilen nicht verändert werden, sie müssen nach Bedarf mit 255 aufgefüllt werden. Außerdem muß jede dieser Zeilen mit mindestens einem 255 enden, da diese Zahl als Ende-Kennzeichen einer Sequenz dient.

Die Zeile 530 legt die Sequenz für einen Carriage-Return Linefeed fest, sie ist 13,255,255 für Drucker mit Auto-Linefeed und 13,10,255 für Drucker ohne Auto-Linefeed.

Zeile 550 stellt den Drucker auf den für Grafik passenden Zeilenvorschub ein. Für den Epson RX-80 ist dies zum Beispiel ESC »3« 23 oder in Zahlen 27,51,23, aufgefüllt mit 255 auf die erforderliche Länge.

In Zeile 580 wird der Grafikmodus (Bit-Image-Mode) für 640 Punkte/Zeile angesteuert. Beispiele:

In Zeile 600 wird wieder der normale Zeilenabstand eingestellt (ESC »2«) und die -2 in Zeile 610 signalisiert für den DATA-Lader das Ende der DATAs.

Bei Schwarzweiß-Bildern, wie die der Diashow, Supergrafik oder natürlich des erweiterten Simons Basic geht's problemlos. Bei Grafikprogrammen, die im Multicolour-Modus arbeiten, treten zwei Schwierigkeiten auf:

1. Da HI-EDDI im High-Resolution-Modus arbeitet, sehen Multicolour-Bilder recht ungewöhnlich aus. Dabei kommt es vor allem auch auf die Farborganisation des betreffenden Programmes an: Die des Koalapainter ist so chaotisch, daß dessen Bilder im »HI-EDDI« fast unbrauchbar sind (Ein und dieselbe Farbe erhält verschiedene Multicolour-Codes). Paint-Magic-Bilder sind dagegen sofort »druckreif«.
2. Für Multicolour-Bilder gibt es kein einheitliches Format für die Speicherung auf Diskette, so daß viele Bilder nicht direkt von »HI-EDDI« geladen werden können.

Laden von Bildern

Für den Koalapainter tritt die 2. Schwierigkeit nicht auf, seine Bilder lassen sich sofort laden. Den Filenamen gibt man dabei in der Form »?PIC...« ein, um das Steuerzeichen am Anfang auszutricksen. Da jedoch Koala-Bilder länger sind als die vom »HI-EDDI«, werden Farbinformationen nicht nur des geladenen, sondern noch eines weiteren Bildes überschrieben. Nach dem Laden eines Koala-Bildes herrscht somit buntes Chaos im Computer, das erst durch Neu-Einfärben beseitigt werden muß.

Etwas komplizierter wird's bei Paint Magic, dessen Bilder sich nicht direkt laden lassen, da sie in Form eines Basic-Laders abgespeichert werden. Man muß erst ein Bild mit »LOAD"..",8« laden, dann mit RUN starten, anschließend »HI-EDDI« laden und in Betriebsart 1 starten. Die 1 ist nötig, damit »HI-EDDI« das im Speicher befindliche Bild nicht löscht. Außerdem darf »HI-EDDI« nicht im Farbbetrieb gestartet werden, da das Paint-Magic-Bild im Speicherbereich $4000-$6000 liegt, wo »HI-EDDI« sonst seine Farbinformationen ablegen würde. Nach dem Start ist das Bild in Speicher Nummer 7 zu finden.

Lädt man übrigens im Schwarzweiß-Betrieb ein Farbbild (36 Blocks), dann wird auch hier durch die Farbinformation ein Teil eines anderen Bildes überschrieben, was sich durch »Gerümpel« in den ersten paar Zeilen dieses Bildes bemerkbar macht.

Bilder in Basic verwenden

Will man »HI-EDDI«-Bilder in normalen Basic-Programmen (oder mit der Simons Basic Modulversion) laden, dann tut's das Listing »Pic-Lader (Listing 3), das natürlich in eigene Programme eingebaut werden kann. Das Bild wird unter das Betriebssystem-ROM gelegt, der Farbspeicher ab $C000 (=49152, wie bei Simons Basic). Damit wird kein Basic-Speicherplatz belegt. Allerdings dürfen nur Schwarzweiß-Bilder (33 Blocks) geladen werden, bei Farbbildern gibt's einen Absturz.

Auch am Programm selbst sind mir noch vier Kleinigkeiten aufgefallen:
1. Bei einem STOP/RESTORE schreibt das Betriebssystem ein paar Bytes in den RAM-Bereich über $E000. Diese sind dann in einem der Bilder am unteren Rand als »Gerümpel« zu sehen.
2. Beim Befehl »Paint« kann es bei extrem verwinkelten Flächen vorkommen, daß Teile übersehen werden (Grund dafür ist der begrenzte Stack des 6510). Diese müssen eben extra aufgefüllt werden.
3. Der Sequenzstring sollte nicht länger als zirka 40 Zeichen werden, sonst könnte es bei LOAD oder SAVE zu einem OUT OF MEMORY ERROR kommen. Sollte das passieren, so muß Hi-Eddi mit GOTO 150 gestartet und der Sequenzstring verkürzt werden.
4. Bei gewaltigem »Herumwürgen« am Joystick kann es vorkommen, daß der Computer meint, es wurde eine Taste gedrückt. Ergebnis ist dann eine Befehlsausführung, die man gar nicht wollte. Allerdings passiert das äußerst selten, vielleicht liegt's auch nur an meinem mickrigen Joystick. Wenn ansonsten mal ein Befehl nicht das tut, was er sollte, dann liegt es meistens an einer eingerasteten SHIFT-LOCK-Taste. Bei Befehlen, die die SHIFT-Taste nicht benutzen, fällt es gar nicht auf. Will man jedoch zum Beispiel den Fore-Modus anwählen und färbt statt dessen das ganze Bild ein, dann war's SHIFT-LOCK!

(Hans Haberl/ah)

```
1 REM******       SAVER       ******          <081>
10 DATA 198,1,169,96,141,123,129,162,2,32,
   201,255,162,97,160,0,185,159,127           <015>
20 DATA 32,210,255,200,202,208,246,169,0,1
   33,253,169,128,133,254,162,64,160          <045>
30 DATA 0,177,253,32,210,255,200,208,248,2
   30,254,202,208,243,185,77,198,32           <008>
40 DATA 210,255,200,208,247,32,204,255,230
   ,1,96,1,8,13,8,191,7,158,40,50             <137>
50 DATA 48,54,51,41,0,0,0,120,169,96,133,2
   52,169,8,133,253,169,0,133,254             <045>
60 DATA 169,128,133,255,162,64,32,69,8,32,
   10,128,169,169,141,123,129,169             <251>
70 DATA 96,133,252,169,72,133,253,169,77,1
   33,254,169,198,133,255,162,1,32            <251>
80 DATA 69,8,76,123,129,160,0,177,252,145,
   254,136,208,249,230,253,230,255            <009>
90 DATA 202,208,242,96,0,0,0,0,0,0,0,0,0,0 <207>
100 POKE 56,112:CLR                           <079>
110 FOR I=32605 TO 32767:READ A:S=S+A         <000>
120 POKE I,A:NEXT I                           <034>
130 IF S<>21569 THEN PRINT"FEHLER IN DATAS
    ":GOTO 200                                <034>
140 INPUT"NAME DER NEUEN VERSION";N$          <254>
150 OPEN 15,8,15:OPEN 2,8,2,N$+",P,W"         <075>
160 GOSUB 170:SYS 32605:CLOSE 2:GOSUB 170:
    CLOSE 15:GOTO 200                         <197>
170 INPUT#15,A,B$,C,D:PRINT A;B$;C;D          <103>
180 IF A>0 THEN CLOSE 2:CLOSE 15:GOTO 200     <156>
190 RETURN                                    <248>
200 POKE 56,128:CLR:END                       <019>

© 64'er
```

Listing 2. So kann Simons Basic erweitert werden

```
30 REM*  LADEN VON HI-EDDI-BILDERN  *          <222>
100 INPUT"FILENAME";N$                        <042>
110 POKE 781,8:REM  GERAETEADRESSE            <091>
120 POKE 782,0:REM  SEKUNDAERADRESSE          <225>
130 SYS 65466:REM  FILEPARAMETER SETZEN       <203>
140 POKE 780,LEN(N$)                          <181>
150 POKE 781,192:POKE 782,2:REM  POINTER A
    UF FILENAMEN                              <063>
160 FOR I=1 TO LEN(N$):POKE 703+I,ASC(MID$
    (N$,I)):NEXT:REM  FILENAME                <239>
170 SYS 65469:REM  NAMENSPARAMETER SETZEN     <018>
180 POKE 780,0:REM  LOAD-FLAG                 <112>
190 POKE 781,0:POKE 782,224:REM  ADRESSE=$
    E000                                      <220>
200 SYS 65493:REM  LOAD                       <083>
210 REM  GRAFIK EINSCHALTEN                   <002>
220 M1=PEEK(56576):POKE 56576,M1 AND 252:R
    EM  VIC-ADRESSBEREICH AB $C000            <212>
230 M2=PEEK(53272):POKE 53272,8:REM  VIDEO
    RAM- UND GRAFIK-ADRESSEN                  <070>
240 M3=PEEK(53265):POKE 53265,M3 OR 32:REM
      HIRES EINSCHALTEN                       <024>
250 F=16*7+6:REM  16*VORDERGRUNDFARBE+HINT
    ERGRUNDFARBE                              <037>
260 FOR I=49152 TO 50152:POKE I,F:NEXT        <138>
270 POKE 198,0:WAIT 198,1:REM  AUF TASTE W
    ARTEN                                     <030>
280 POKE 56576,M1:POKE 53272,M2:POKE 53265
    ,M3:REM  WIEDER NORMALZUSTAND             <030>

© 64'er
```

Listing 3. HI-EDDI-Bilder in eigenen Programmen verwenden

Hardcopy von HI-EDDI auf dem Plotter VC 1520

Mit diesem Programm ist es möglich, Hardcopies von HI-EDDI mit dem Printer-Plotter VC 1520 zu erstellen.

Geben Sie zuerst »Copy 1520« (Listing 1) mit dem MSE ein und speichern Sie es ab. Jetzt muß noch das HI-EDDI Steuerprogramm geändert werden. Bei dem Steuerprogramm (Listing 2) sind die Zeilen 600 bis 750 zu ändern.

Die Hardcopy kann jetzt von HI-EDDI per Tastendruck oder Menü nachgeladen werden. Es ist möglich, mehrere Bilder (nahtlos) untereinander oder übereinander (Overlay) zu drucken.

(Stephan Paetzold/ah)

```
50 IF A<2 THEN A=A+2:LOAD"HI-EXE",8,1               <107>
60 IF A=3 THEN 150                                   <027>
70 IF A=4 THEN 650                                   <200>
100 INPUT"{CLR}BETRIEBSART";C:POKE 780,C:P
    OKE 56,13:CLR:W$="123456":SYS 3328              <254>
110 IF PEEK(8073)AND 64 THEN A=3:LOAD"MENU
    E",8,1                                           <087>
150 SYS 3337:A=PEEK(8064)                            <119>
160 ON A+1 GOSUB 200,250,300,290,400,600,2
    90,290,500,510,300:GOTO 150                      <254>
200 OPEN 3,8,0,"$0":GET#3,C$,C$                      <154>
210 GET#3,C$,C$,L$,H$:IF C$=""THEN CLOSE 3
    :WAIT 198,1:RETURN                               <094>
220 PRINT 256*ASC(H$+CHR$(0))+ASC(L$+CHR$(
    0));:SYS 3334:GOTO 210                           <018>
250 PRINT"{CLR,DOWN}SEQUENZ:":PRINT"{DOWN,
    2SPACE}"W$:INPUT"{HOME,3DOWN}";W$:IF L
    EN(W$)<2 THEN 250                                <062>
290 RETURN                                           <094>
300 POKE 8115,8:PRINT"{CLR,DOWN}G=GRAFIKBI
    LD":IF PEEK(8073)>127 THEN PRINT"F=FAR
    BBILD"                                           <036>
310 PRINT"S=SPRITE":PRINT"Z=ZEICHENSATZ":P
    RINT"{DOWN,SPACE}-->";                           <017>
320 GET C$:IF C$<>"G"AND C$<>"F"AND C$<>"S
    "AND C$<>"Z"THEN 320                             <240>
330 PRINT C$:POKE 780,ASC(C$):H$=W$:INPUT"
    {DOWN}FILENAME";W$:SYS 3331:W$=H$:H$="
    "                                                <046>
340 OPEN 1,8,15,H$:INPUT#1,A,H$:PRINT A;H$
    :IF A>0 THEN WAIT 198,1                          <073>
350 CLOSE 1:RETURN                                   <223>
400 POKE 631,34:POKE 198,1:H$="":INPUT"{CL
    R,DOWN}";H$:GOTO 340                             <207>
500 A=8075:GOTO 520                                  <053>
510 A=8079                                           <140>
520 D=PEEK(8074):C=PEEK(A+D):PRINT"{CLR,DO
    WN}F"2*D+1:PRINT"SCHRITTWEITE :"C,               <055>
530 INPUT C:IF C<1 OR C>160 THEN 530                 <066>
540 POKE A+D,C:RETURN                                <189>
600 IF A=5 THEN A=4:LOAD"COPY 1520",8,1              <045>
650 INPUT"{CLR}BILDNUMMER";D:C=(D AND 7)             <212>
660 INPUT"FARBE (0-3)";B$:POKE 1000,0:POKE
     1001,VAL(B$)                                    <099>
670 FOR I=0 TO 6                                     <103>
680 IF PEEK(8086+I)=D THEN LET Z=(I+1)*819
    2                                                <162>
690 NEXT I                                           <010>
700 X=INT(Z/256):POKE 3399,X:POKE 3392,Z-X
    *256                                             <025>
710 POKE 780,C:SYS 3328                              <028>
720 PRINT"OVERLAY";:GOSUB 800:IF C$="J"THE
    N 630                                            <159>
730 PRINT"NAHTLOS";:GOSUB 800:IF C$="N"THE
    N LET A=1:GOTO 50                                <161>
740 OPEN 1,6,1:PRINT#1,"M",0,-180:CLOSE 1            <106>
750 OPEN 1,6,0:PRINT#1," ":CLOSE 1:GOTO 65
    0                                                <077>
800 PRINT" ? (J/N)";                                 <205>
810 GET C$:IF C$<>"J"AND C$<>"N"THEN 810             <210>
820 PRINT C$:RETURN                                  <230>
© 64'er
```

Listing 2. »Steuerprogramm HI-EDDI«. Bitte beachten Sie die Eingabehinweise auf Seite 16.

```
programm : copy 1520      0d00 103b
-------------------------------------

0d00 : a2 00 8e 0e dc 78 86 b7   23
0d08 : e8 86 b8 86 b9 a2 06 86   08
0d10 : ba 20 c0 ff a2 02 86 b8   d0
0d18 : 86 b9 a2 00 86 b7 a2 06   e0
0d20 : 86 ba 20 c0 ff a2 00 8e   56
0d28 : 40 03 8e 41 03 8e 43 03   6d
0d30 : 8e 46 03 8e 47 03 8e 48   cb
0d38 : 03 8e 49 03 8e 4c 03 a2   d2
0d40 : 00 86 f7 8e 94 0e a2 e0   59
0d48 : 86 f8 8e 98 0e a2 01 8e   18
0d50 : 4b 03 a2 d0 8e a3 0d 8e   37
0d58 : d7 0d a2 02 20 c9 ff ad   4a
0d60 : e9 03 18 69 30 20 d2 ff   4d
0d68 : a9 0d 20 d2 ff 20 cc ff   2e
0d70 : a9 34 85 01 a0 00 b1 f7   76
0d78 : a2 37 86 01 48 18 a5 f7   43
0d80 : 69 08 90 02 e6 f8 85 f7   8e
0d88 : ad e8 03 f0 38 68 a2 00   da
0d90 : 8e 4a 03 0a 2e 4a 03 0a   9b
0d98 : 2e 4a 03 48 aa ae 4a 03   04
0da0 : ec 4b 03 d0 03 20 00 0f   5c
0da8 : ee 49 03 ee 41 03 d0 03   4f
0db0 : ee 40 03 ee 41 03 d0 03   d2
0db8 : ee 40 03 ad 49 03 c9 04   19
0dc0 : d0 cb 4c ee 0d 68 a2 00   05
0dc8 : 8e 4a 03 18 0a 2e 4a 03   80
0dd0 : 48 ad 4a 03 cd 4b 03 d0   c7
0dd8 : 03 20 00 0f ee 49 03 ee   f0
0de0 : 41 03 d0 03 ee 40 03 ad   90
0de8 : 49 03 c9 08 d0 d7 a2 00   7d
0df0 : 8e 49 03 68 ee 48 03 ad   89
0df8 : 48 03 c9 28 b0 03 4c 70   6e
0e00 : 0d ad 4c 03 f0 03 20 00   ff
0e08 : 0f ee 43 03 ad 4e 03 8d   34
0e10 : 41 03 ad 4d 03 8d 40 03   8b
0e18 : 20 00 0f a2 00 8e 48 03   ec
0e20 : 8e 40 03 8e 41 03 8e 4c   60
0e28 : 03 a9 d0 8d a3 0d 8d d7   6e
0e30 : 0d c6 f8 a5 f7 38 e9 3f   fa
0e38 : b0 02 c6 f8 85 f7 ee 47   1c
0e40 : 03 ad 47 03 c9 08 f0 03   f3
0e48 : 4c 70 0d a2 00 8e 47 03   fc
0e50 : e6 f8 a5 f7 18 69 38 90   e9
0e58 : 02 e6 f8 85 f7 ee 46 03   d2
0e60 : ad 46 03 c9 19 f0 03 4c   e8
0e68 : 70 0d 20 cc ff ad e8 03   17
0e70 : d0 03 4c f6 0e ee 4b 03   3f
0e78 : ad 4b 03 c9 04 f0 77 a2   b0
0e80 : 00 8e 46 03 8e 43 03 8e   e6
0e88 : 40 03 8e 41 03 8e 4d 03   f5
0e90 : 8e 4e 03 a2 00 86 f7 a2   b4
0e98 : e0 86 f8 a2 01 20 c9 ff   86
0ea0 : a2 00 8a 48 bd e7 0e 20   82
0ea8 : d2 ff 68 aa e8 e0 09 d0   45
0eb0 : f1 a2 01 20 c9 ff a2 00   5e
0eb8 : 8a 48 bd f0 0e 20 d2 ff   21
0ec0 : 68 aa e8 e0 06 d0 f1 a2   c7
0ec8 : 02 20 c9 ff ad 4b 03 38   fe
0ed0 : e9 01 a8 b9 e9 03 18 69   85
0ed8 : 30 20 d2 ff a9 0d 20 d2   f6
0ee0 : ff 20 cc ff 4c 70 0d 4d   39
0ee8 : 2c 30 2c 2d 32 30 31 0d   61
0ef0 : 4d 2c 30 2c 30 0d 20 2f   2f
0ef8 : f3 a9 01 8d 0e dc 58 60   9c
0f00 : ad 40 03 48 ad 41 03 48   19
0f08 : ad 43 03 48 a2 00 8e 43   0c
0f10 : 03 8e 44 03 8e 45 03 c9   7f
0f18 : 64 90 08 e9 64 ee 43 03   d4
0f20 : 4c 17 0f c9 0a 90 08 e9   0e
0f28 : 0a ee 44 03 4c 23 0f c9   c9
0f30 : 01 90 08 e9 01 ee 45 03   5b
0f38 : 4c 2f 0f ad 41 03 ae 40   fd
0f40 : 03 f0 11 a2 02 8e 40 03   f0
0f48 : a2 05 8e 41 03 e8 8e 42   6f
0f50 : 03 4c 5f 0f a2 00 8e 40   18
0f58 : 03 8e 41 03 8e 42 03 c9   ee
0f60 : 64 90 08 ee 40 03 e9 64   79
0f68 : 4c 5f 0f c9 0a 90 08 ee   84
0f70 : 41 03 e9 0a 4c 6b 0f c9   de
0f78 : 01 90 08 ee 42 03 e9 01   87
0f80 : 4c 77 0f ad 41 03 18 69   61
0f88 : 08 8d 41 03 ad 42 03 c9   94
0f90 : 0a 90 08 e9 0a 8d 42 03   3e
0f98 : ee 41 03 ad 41 03 c9 0a   05
0fa0 : 90 0b e9 0a 8d 41 03 ee   3e
0fa8 : 40 03 4c 9b 0f a2 01 20   3b
0fb0 : c9 ff ad 4c 03 f0 08 a9   99
0fb8 : 44 20 d2 ff 4c c4 0f a9   3b
0fc0 : 4d 20 d2 ff a9 2c 20 d2   f4
0fc8 : ff ad 40 03 18 69 30 20   dc
0fd0 : d2 ff ad 41 03 18 69 30   2c
0fd8 : 20 d2 ff ad 42 03 18 69   86
0fe0 : 30 20 d2 ff a9 2c 20 d2   f7
0fe8 : ff a9 2d 20 d2 ff ad 43   76
0ff0 : 03 18 69 30 20 d2 ff ad   54
0ff8 : 44 03 18 69 30 20 d2 ff   40
1000 : ad 45 03 18 69 30 20 d2   52
1008 : ff a9 0d 20 d2 ff 68 8d   0d
1010 : 43 03 68 8d 41 03 68 8d   8a
1018 : 40 03 ad a3 0d 49 20 8d   70
1020 : a3 0d 8d d7 0d ad 4c 03   1e
1028 : 49 01 8d 4c 03 ad 41 03   87
1030 : 8d 4e 03 ad 40 03 8d 4d   48
1038 : 03 60 34 ff ff ff ff ff   78
```

Listing 1. »Copy 1520«. Bitte beachten Sie bei der Eingabe den MSE.

HI-EDDI mit Itoh 8510

Besitzern des Itoh 8510 bietet dieses Programm die Möglichkeit, von HI-EDDI Hardcopies zu erzeugen.

Mit HI-EDDI lassen sich mit minimaler Hardware-Ausrüstung (C 64, Diskettenlaufwerk 1541, Joystick) sehr komfortable Grafiken erzeugen. Doch was nützt die schönste Grafik auf dem Bildschirm, wenn man sie nicht ausdrucken kann?

Die Druckroutine HI-PRINT arbeitet leider nur mit Epson-Druckern zusammen. Da der Itoh 8510 ein verbreiteter und guter Drucker ist, wurden für ihn die im folgenden beschriebenen Programme erstellt. Er wird über eine Centronics-Schnittstelle via User-Port am C 64 angeschlossen.

Zusätzlich wurden einige Features eingebaut, die die Epson-Hardcopy-Routine nicht besitzt. So wäre die Vorgabe von Leerspalten vom linken Rand aus (falls noch Platz ist) eine nützliche Sache. Um zwei Grafiken in der Horizontalen aneinander zu setzen, wurde dazu die volle Bildschirmbreite in 40 Spalten zu je 8 (40 x 8 = 320) Punkten (wie im Textmodus) organisiert. Um jetzt zwei Grafiken nebeneinander zu plazieren, kann dazu die Startspalte (wird mitgedruckt) der links stehenden Grafik (Grafik Nr. 1) und die Endspalte (wird nicht mitgedruckt) der rechts stehenden Grafik (Grafik Nr. 2) vorgewählt werden.

Die Ausdruckbreite wird dazu in drei Sektoren von links nach rechts aufgeteilt.

0. Sektor: Abstand (SPC) in Punkten vom linken Papierrand.
1. Sektor: Grafik Nr. 1 (Bild-Nr. ; Startspalte)
   Bild-Nr. = 0 = > kein Bild, Startspalte = 40
   Bild wird rechtsbündig gedruckt
   0 < = Startspalte < = 39
   Startspalte = 0 = > volle Hardcopy von Grafik Nr. 1
2. Sektor: Grafik Nr. 2 (Bild-Nr. ; Endspalte)
   Bild-Nr. = 0 = > kein Bild, Endspalte = 0
   Bild wird linksbündig gedruckt
   1 < = Endspalte < = 40
   Endspalte = 40 = > volle Hardcopy von Grafik Nr. 2

Insgesamt kann folgende Formel für die Aufteilung der drei Sektoren angegeben werden:

SPC + (40—Startspalte) * 8 + Endspalte * 8 < = 640

Im Steuerprogramm werden nacheinander die Bild-Nr. von Sektor 1, dann (falls Bild-Nr. < > 0) die Startspalte von Grafik-Nr. 1, dann die Bild-Nr. von Sektor 2 und (falls Bild-Nr. < > 0) die Endspalte von Grafik-Nr. 2, und schließlich die Anzahl der Leerspalten vor Grafik-Nr. 1 abgefragt.

Die Maximalwerte werden stets durch das Steuerprogramm vorgegeben, so daß in vielen Fällen ein einfaches RETURN genügt.

Um die Druckroutine »HI-P(ITOH8510)« für den Itho 8510 in das Steuerprogramm HI-EDDI einzubinden, mußte dieses auf die Bedürfnisse der Druckroutine angepaßt werden und heißt nun »HI-E(ITOH8510)«.

Dazu wurden (aus Speicherplatzgründen) alle Zeilen neu numeriert und die Textausgabepassagen verkürzt. In den Programmzeilen 24 und folgende befindet sich das Drucksteuerprogramm.

Es wird hierbei dringend davor gewarnt, das neue Steuerprogramm HI-E(ITOH8510) zu erweitern oder zu ändern, da sonst das Steuerprogramm (inklusive Variablen) nicht mehr in den verfügbaren Basic-Speicherplatz paßt. Dies gilt auch für den eingebauten Parametercheck, da in der Druckroutine selber keine Plausibilitätsprüfung mehr vorgenommen wird.

Für die Initialisierung des Druckers und die Ausgabe der Hardcopy sollte sich der Drucker im SELECT-Modus befinden, da sonst das Programm hängenbleibt (Acknowledge-Signal wird erwartet).

Falls aus irgendeinem Grunde (Drucker beziehungsweise Kabel defekt, falsch oder gar nicht angeschlossen) dieser Fall eintritt, kann mit RUN/STOP-RESTORE die Druckroutine beendet werden. HI-EDDI kann dann wieder mit RUN gestartet werden. (Das RUN ist notwendig, um HI-EXE nachzuladen, da aus Speicherplatzgründen nach der Anwahl des Drucksteuerprogramms immer die Druckroutine über HI-EXE nachgeladen wird.)

Der Drucker sollte mittels der DIP-Schalter auf CR = CR und 8-Bit-Data eingestellt sein. Da am Anfang kein Zeilenvorschub ausgegeben wird, sollte man dafür sorgen, daß der Druckkopf sich auf einer freien Zeile befindet. Außerdem sollte das Papier zentriert werden, da sich der Ausdruck über die volle Papierbreite erstrecken kann.

Die Anpassung des Itoh 8510A an HI-EDDI besteht im wesentlichen aus zwei Schritten:

```
1 IF A<2 THEN A=A+2:LOAD"HI-EXE",8,1                <058>
2 IF A=3 THEN 6                                     <207>
3 IF A=4 THEN 25                                    <035>
4 INPUT"B-ART";C:POKE 780,C:POKE 56,13:CLR
  :W$="123456":SYS 3328                             <117>
5 IF PEEK(8073)AND 64 THEN A=3:LOAD"MENUE"
  ,8,1                                              <238>
6 SYS 3337:A=PEEK(8064):ON A+1 GOSUB 7,10,
  12,11,18,24,11,11,19,20,12:GOTO 6                 <059>
7 OPEN 3,8,0,"$0":GET#3,C$,C$                       <217>
8 GET#3,C$,C$,L$,H$:IF C$=""THEN CLOSE 3:W
  AIT 198,1:RETURN                                  <148>
9 PRINT 256*ASC(H$+CHR$(0))+ASC(L$+CHR$(0)
  );:SYS 3334:GOTO 8                                <156>
10 PRINT"{CLR,DOWN}SEQ.:":PRINT"{DOWN,2SPA
   CE}"W$:INPUT"{HOME,3DOWN}";W$:IF LEN(W$
   )<2 THEN 10                                      <096>
11 RETURN                                           <069>
12 POKE 8115,8:PRINT"{CLR,DOWN,RVSON}G{RVO
   FF}-BILD":IF PEEK(8073)>127 THEN PRINT"
   {RVSON}F{RVOFF}-BILD"                            <152>
13 PRINT"{RVSON}S{RVOFF}PRITE":PRINT"{RVSO
   N}Z{RVOFF}-SATZ":PRINT"{DOWN}->";                <208>
14 GET C$:IF C$<>"G"AND C$<>"F"AND C$<>"S"
   AND C$<>"Z"THEN 14                               <093>
15 PRINT C$:POKE 780,ASC(C$):H$=W$:INPUT"{
   DOWN}F-NAME";W$:SYS 3331:W$=H$:H$=""             <213>
16 OPEN 1,8,15,H$:INPUT#1,A,H$:PRINT A;H$:
   IF A>0 THEN WAIT 198,1                           <003>
17 CLOSE 1:RETURN                                   <144>
18 POKE 631,34:POKE 198,1:H$="":INPUT"{CLR
   ,DOWN}";H$:GOTO 16                               <210>
19 A=8075:GOTO 21                                   <059>
20 A=8079                                           <160>
21 D=PEEK(8074):C=PEEK(A+D):PRINT"{CLR,DOW
   N}F"2*D+1:PRINT"S-WEITE:"C,                      <165>
22 INPUT C:IF C<1 OR C>160 THEN 22                  <125>
23 POKE A+D,C:RETURN                                <180>
24 A=4:LOAD"HI-P(ITOH8510)",8,1                     <215>
25 INPUT"{CLR,DOWN}S1:B-NR.(0=KB)";C:POKE
   3415,C AND 7:IF C=0 THEN D=40:GOTO 27            <236>
26 INPUT"SSP.(<=39;0=VOLL){2SPACE}0{3LEFT}
   ";D:IF D<0 OR D>39 THEN 26                       <222>
27 POKE 3388,D:C=40+D:IF C>40 THEN C=40             <119>
28 INPUT"{DOWN}S2:B-NR.(0=KB)";D:POKE 3435
   ,D AND 7:IF D=0 THEN 30                          <041>
29 PRINT"ESP.(>0;<=";C;"{LEFT}) ";C;:INPUT
   "{5LEFT}";D:IF D<1 OR D>C THEN 29                <191>
30 POKE 3451,D:C=640-(40-PEEK(3388)+D)*8            <024>
31 PRINT"{DOWN}SPC(=>0;<=";C;"{LEFT}){2SPA
   CE}";C;:INPUT"{6LEFT}";D:IF D<0 OR D>C
   THEN 31                                          <194>
32 C=INT(D/256):POKE 3383,C:INPUT"{DOWN}DR
   UCK{2SPACE}J{3LEFT}";C$:IF C$<>"J"THEN
   34                                               <106>
33 POKE 3381,D-C*256:SYS 3328:INPUT"{DOWN}
   NOCHMAL{2SPACE}J{3LEFT}";C$:IF C$="J"TH
   EN 25                                            <058>
34 A=1:GOTO 1                                       <016>
```

Listing 1. »HI-E(ITOH8510)«

```
10 REM #### PROGRAMM ZUR ERZEUGUNG #####   <190>
20 REM ## VON HI-P(ITOH8510) AUF DISK ##   <124>
30 REM AUTOR: F.ZIESCHE' AM: 13. 2.85 ##   <068>
40 RESTORE:PS=0:REM DATA'S UEBERPRUEFEN    <136>
50 PRINT"{CLR,DOWN,SPACE}DATA-ZEILEN WERDE
   N UEBERPRUFT !"                         <100>
60 READ BY : IF BY < 0 THEN 80             <145>
70 PS = PS + BY : GOTO 60                  <016>
80 IF PS+BY <> 0 THEN PRINT"FEHLER IN DATA
   -ZEILEN !":LIST 160-                    <036>
90 INPUT"{DOWN,SPACE}DISKETTE EINGELEGT{3S
   PACE}JA{4LEFT}";IN$:IF IN$<>"JA"THEN 90 <159>
100 RESTORE : OPEN 3,8,3,"HI-P(ITOH8510),P
    ,W"                                    <061>
110 PRINT#3,CHR$(0);CHR$(13);:REM STARTADR
    ESSE                                   <115>
120 READ BY : IF BY < 0 THEN 140           <233>
130 PRINT#3,CHR$(BY); : GOTO 120           <044>
140 CLOSE 3 : PRINT"{DOWN,SPACE}PROGRAMM W
    URDE ERZEUGT !"                        <107>
150 END                                    <152>
160 REM DATA-ZEILEN FUER HI-P(ITOH8510)    <018>
170 DATA 165,1,72,169,55,133,1,173,87,13,3
    2,186,13,141,170,14,173,107,13,32      <217>
180 DATA 186,13,141,223,14,169,0,141,164,1
    4,141,217,14,141,174,13,32,248,13      <145>
190 DATA 32,212,13,169,27,32,111,14,169,70
    ,32,111,14,169,0,160,0,32,47,14        <162>
200 DATA 169,0,141,148,14,169,0,141,201,14
    ,169,27,32,111,14,169,83,32,111        <055>
210 DATA 14,169,0,160,0,32,47,14,169,0,240
    ,16,32,135,14,32,241,14,238,148        <007>
220 DATA 14,173,148,14,201,40,144,240,169,
    0,240,16,32,188,14,32,241,14,238       <092>
230 DATA 201,14,173,201,14,201,0,144,240,2
    4,173,164,14,105,64,141,164,14,173     <187>
240 DATA 170,14,105,1,141,170,14,24,173,21
    7,14,105,64,141,217,14,173,223,14      <185>
250 DATA 105,1,141,223,14,169,13,32,111,14
    ,169,10,32,111,14,238,174,13,169       <197>
260 DATA 0,201,25,176,3,76,42,13,104,133,1
    ,96,162,7,221,149,31,240,4,202,208     <247>
270 DATA 248,232,138,10,10,10,10,10,96,0,0
    ,0,0,0,0,0,0,24,173,123,13,105,40      <020>
280 DATA 56,237,60,13,133,251,169,0,133,25
    2,160,3,6,251,38,252,136,208,249       <132>
290 DATA 165,251,164,252,141,80,13,140,82,
    13,96,169,255,141,3,221,173,2,221      <055>
300 DATA 9,4,141,2,221,173,0,221,9,4,141,0
    ,221,162,0,189,27,14,32,111,14,232     <033>
310 DATA 224,20,144,245,96,127,27,93,27,78
    ,27,84,49,54,27,102,27,76,48,48        <183>
320 DATA 48,27,48,27,62,133,99,132,98,56,1
    62,144,32,73,188,32,221,189,162        <085>
330 DATA 1,189,0,1,240,6,232,224,6,144,246
    ,202,202,142,81,14,56,169,4,233        <181>
340 DATA 0,240,9,168,169,48,32,111,14,136,
    208,248,172,81,14,240,12,162,1,189     <114>
350 DATA 0,1,32,111,14,232,136,208,246,96,
    141,1,221,173,0,221,41,251,141,0       <177>
360 DATA 221,9,4,141,0,221,173,13,221,41,1
    6,240,249,96,120,165,1,72,41,248       <033>
370 DATA 133,1,169,0,133,252,169,0,133,251
    ,160,3,6,251,38,252,136,208,249        <080>
380 DATA 24,165,251,105,0,133,251,165,252,
    105,0,133,252,177,251,153,204,13       <176>
390 DATA 200,192,8,144,246,104,133,1,88,96
    ,120,165,1,72,41,248,133,1,169,0       <202>
400 DATA 133,252,169,0,133,251,160,3,6,251
    ,38,252,136,208,249,24,165,251,105     <237>
410 DATA 0,133,251,165,252,105,0,133,252,1
    77,251,153,204,13,200,192,8,144        <236>
420 DATA 246,104,133,1,88,96,160,128,140,4
    ,15,169,0,141,8,15,140,10,15,162       <203>
430 DATA 7,189,204,13,41,0,240,7,169,0,9,0
    ,141,8,15,78,10,15,202,16,236,173      <206>
440 DATA 8,15,32,111,14,78,4,15,144,215,96 <148>
450 DATA -57076 :REM  PRUEFSUMME           <184>
```


Listing 2. »HI-P(ITOH8510).L«
Bitte beachten Sie die Eingabehinweise auf Seite 16.

1. HI-P(ITOH8510).L (Listing 2) abtippen und starten. Dieses ist ein Ladeprogramm, das die Druckroutine HI-P(ITOH 8510) auf Diskette ablegt. Es wird (nach erfolgreichem Test) zum Arbeiten mit HI-EDDI nicht mehr benötigt.

2. HI-E(ITOH8510) (Listing 1.) abtippen und speichern. Es ist das angepaßte Steuerprogramm und ersetzt das alte HI-EDDI.

So, das wär's eigentlich. HI-EDDI mit Itoh 8510A ist nun lauffähig. Mir bleibt nur noch, viel Erfolg mit HI-EDDI zu wünschen.

(Frank Ziesche/ah)

HI-EDDI mit der Datasette

Die Anpassung von HI-EDDI an die Datasette war doch nicht ganz so einfach wie zuerst gedacht. Hier sind die notwendigen Informationen.

Die folgenden Listingshinweise beziehen sich auf Seite 58ff.

a) Am Listing 3 (HI-EDDI) folgende Änderungen vornehmen: In Zeile 50, 110 und 640 LOAD" .. ",8,1 durch LOAD" .. ",1 ersetzen.

```
160  ON A+1 GOSUB 290,250,300,290,290,600,290,290,500,
     510,300:GOTO150
300  POKE8115,1:PRINT...
330  PRINT C$:POKE 780,ASC(C$):H$=W$:W$=" "
340  INPUT"{Cursor Down}FILENAME";W$:SYS
     3331:W$=H$:H$=" "
350  RETURN
```

Folgende Zeilen weglassen: 200-220, 400

b) Das so modifizierte Listing 3 als erstes auf die Programmkassette aufnehmen.

c) Folgende Befehle im Direktmodus eingeben:
NEW:POKE8192,0:POKE43,1:POKE44,32:NEW

d) Programmkassette einlegen (nach Listing 3) und Listing 2 starten. Wenn keine Fehler mehr drin sind, erzeugt der Lader auf der Kassette das Maschinenprogramm HI-EXE.

e) HI-EDDI ist jetzt bereits lauffähig: Programmkassette zurückspulen, Computer kaltstarten, Listing 3 laden und starten. Das Maschinenprogramm wird nun geladen.

f) Die HI-EDDI-Befehle LOAD und SAVE funktionieren ebenfalls, somit kann auch schon das Menü erstellt werden. Die fertige Menütafel (mit Listing 4 verbunden) sollte an dritter Stelle auf die Programmkassette aufgenommen werden.

g) Hier die erforderlichen Änderungen am Listing 1: Zunächst wieder die POKEs wie unter c) eingeben und dann Listing 1 laden und folgende Änderungen vornehmen:

```
720  FOR I=3326 TO 3708
730  READA:IF A<0THEN 730
740  POKEI,A:NEXT
750  POKE43,0:POKE44,13:POKE45,125:POKE46,14
760  SAVE"HI-PRINT",1,1
770  END
```

h) Programmkassette einlegen (an 4. Stelle) und Listing 1 starten. Auf der Kassette wird nun das Maschinenprogramm HI-PRINT erzeugt.

Wenn nun im Programmlauf vor einem Druckvorgang diese Druckroutine geladen wurde, muß die Kassette zurückgespult werden, so daß das Maschinenprogramm HI-EXE nach Beendigung des Druckvorganges wieder geladen werden kann.

(Hans Haberl/ah)

Trickfilm mit dem C 64

In die vierte Dimension, die bewegte dreidimensionale Grafik, dringen Sie mit diesem Programm vor. Sie können so mit einfachsten Mitteln Trickfilme mit verblüffenden Eigenschaften erstellen.

Die Unterschiede gegenüber anderen Trickfilmgeneratoren:
— Es muß nur ein Bild eingegeben werden, alle anderen Bilder berechnet der Computer aus diesem Bild.
— Auch komplizierte Bewegungsabläufe können mühelos erzeugt werden (zum Beispiel: um einen Körper um die Z- und X-Achse zu drehen, dabei heranzuholen und nach links zu bewegen, bis er den Bildschirm verläßt, braucht man nur eine DATA-Zeile!).
— Es können sehr viele (bis zu 255) und große Bilder gespeichert werden, da nur die Linienkoordinaten gespeichert werden.
— Es werden keine Basic-Erweiterungen benötigt.

Der »3D-Movie-Maker« muß in zwei Teilen eingegeben werden: zuerst das Maschinenprogramm (Listing 1), dann der Basic-Teil. Wenn das Maschinenprogramm eingegeben und mit SYS 49152 gestartet wurde, so speichert es sich, falls kein Prüfsummenfehler auftrat, selbst als »TRICK.OBJ« ab. Man braucht sich nun nicht mehr darum zu kümmern. Bei dem Basic-Teil (Listing 2), der nun eingegeben werden kann, dürfen alle REM-Zeilen ersatzlos wegfallen. Beim Start des Basic-Teils wird automatisch der Maschinenteil, der sich auf Diskette befinden sollte, nachgeladen.

Bedienung — Eingabe des Körpers

Vor der Benutzung des Programmes, müssen die Punkte Verbindungs- und Bewegungsvorschrift des Körpers in den DATA-Zeilen ab 8000 festgelegt werden. Dies ist notwendig, damit bei Änderungen oder einem erneuten Start nicht alles wieder neu eingegeben werden muß. Die DATA-Zeilen, die das Listing momentan enthält, erzeugen den auf dem Bildschirm herumfliegenden Schriftzug »64'er«. Die Punkte werden mit X-, Y- und Z-Koordinaten eingegeben. Als Endmarke dient hier dreimal die 1000. Bei der Verbindungsvorschrift wird jeweils der Anfangs- und Endpunkt angegeben zum Beispiel: Von Punkt 1 nach Punkt 2 = DATA 1,2. Hier dient zweimal die 1000 als Endmarkierung.

Beispiel: Man will ein Kreuz erzeugen.
Punkte: DATA 0,10,0,10,0,0,—10,0,0
DATA —10, 0,0, 1000,1000,1000
Verbindungsvorschrift: DATA 1,3,2,4,1000,1000

Bei der Bewegungsvorschrift ist es allerdings etwas komplizierter. Hier müssen zweimal drei Verschiebungsarten, drei Drehungsarten und die Dauer des Vorgangs angegeben werden. Zuerst kommt die erste Verschiebung in X-, Y- und Z-Richtung, dann die Drehung um die drei Achsen, nun die zweite Verschiebung und zum Schluß die Dauer des Ganzen. Diese Reihenfolge hat folgenden Sinn:

Wird zuerst verschoben und dann gedreht, dreht sich die Figur um den Bildschirmmittelpunkt. Bei umgekehrter Reihenfolge dreht sich die Figur an beliebiger Stelle um ihren eigenen Mittelpunkt. Hier ein Beispiel einer DATA-Zeile:
DATA 0,0,0,5,0,0, 0,5,0, 10

Die Figur bewegt sich 10 Bilder lang um jeweils 5 Stellen nach oben und dreht sich dabei um jeweils fünf Grad um die X-Achse. Einen Zoomeffekt erreicht man durch Verschieben in der Z-Achse, so kann man zum Beispiel mit DATA 0,0,0,0,0,0,0,0,1,50 eine Figur langsam vergrößern. Hier bildet einmal die 1000 die Schlußmarkierung. Um die vielen verschiedenen Möglichkeiten der Bewegung zu entdecken, lohnt es sich, die verschiedenen Kombinationen auszuprobieren (und dabei mit einfachen Bewegungen anzufangen).

Bedienung — Ablauf des Programms

Der »3D-Movie-Maker« stellt vier Menüpunkte zur Wahl:
1. Erzeugen einer Grafik

Hierbei berechnet der Computer die Anzahl der Bilder und fragt diese noch einmal ab. Sollen alle Bilder gezeichnet werden, braucht nur RETURN eingegeben werden. Nun wird der Speicherbedarf berechnet; reicht der vorhandene Speicher von 23 KByte aus, so beginnt der Rechenvorgang. Dabei wird die Anzahl der fertigen Bilder angezeigt. Ist der Rechenvorgang beendet (bei 150 Bildern mit je 10 Punkten zirka eine halbe Stunde) kehrt das Programm ins Menü zurück. Durch das Drücken der Leertaste kann man sich während der Berechnungen den Film ansehen und dann durch nochmaliges Betätigen fortfahren.

2./3. Laden/ Speichern

Da das Erstellen eines Films recht lange dauert, kann man fertige Filme speichern und laden. Das Programm hängt an den Filenamen automatisch ein ».GRA« an, so daß die Dateien im Directory sofort erkennbar sind. Tritt ein Disk-Error beim Laden oder Speichern auf, landet man wieder in der »INPUT FILENAME«-Zeile. Hier noch ein Hinweis zur LOAD-Routine: Da das Einlesen von 23 KByte mit einer GET-Schleife nicht einwandfrei funktionierte und 15 Minuten dauerte, verwendet die jetzige Version die LOAD-Routine des Betriebssystems.

4. Abspielen

Beim Aufruf dieser Funktion wird man zuerst nach der Anzahl der Durchläufe (maximal 255) und nach der Anzahl der Bilder je Lauf gefragt. Beim letzteren braucht man nur RETURN eingeben, um alle Bilder ablaufen zu lassen.

Der Algorithmus

Das eigentliche Kernstück des Programms ist ein Algorithmus, der die Punkte um einen Winkel dreht, die X-, Y- und Z-Koordinaten in Bildschirmkoordinaten umrechnet und dann die außerhalb liegenden Teile der Linien wegstreicht. Beim Drehen (hier um die Z-Achse) werden erst X- und Y-Koordinaten in Radius und Winkel (im Bogenmaß) umgerechnet; bezogen auf den Koordinatenursprung (siehe Bild 1). Nun wird der Winkel, um den gedreht werden soll, addiert und die neuen Koordinaten werden aus dem neuen Winkel und dem Radius errechnet (siehe Bild 2).

Beim Umrechnen in Bildschirmkoordinaten wird einfach die Z-Koordinate auf die X- und Y-Koordinate so aufgerechnet, daß Punkte, die weiter vorne liegen, vom Mittelpunkt wegrücken. Außerdem wird der Koordinatenursprung in die Bildschirmmitte verlegt. Durch dieses Verfahren wird die Figur mit einem Fluchtpunkt in der Mitte des Bildschirms dargestellt. Das nun folgende Wegstreichen der außerhalb liegenden Linien und Li-

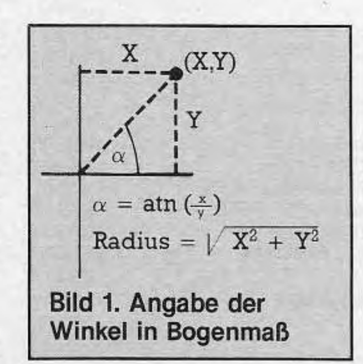


Bild 1. Angabe der Winkel in Bogenmaß

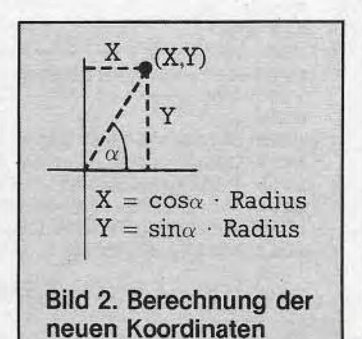


Bild 2. Berechnung der neuen Koordinaten

nienstücke, geschieht mit Hilfe einer Geradengleichung. Falls nur Teile der Geraden außerhalb des Bildschirms liegen, wird der äußerste Wert für X beziehungsweise Y angenommen und die fehlende Koordinate errechnet.

Beschreibung der Maschinenroutine

Die Zeichenroutine des »3D-Movie-Makers« übernimmt die Verwaltung der Grafik und ist, zum Erreichen eines Trickfilmeffekts, ganz auf Geschwindigkeit ausgelegt. Aus diesem Grund erhielt sie die folgenden Merkmale:

— Sie ist in Assembler geschrieben.
— Sie berechnet keine Punkte, sondern zeichnet nur.
— Die Koordinaten werden nicht geprüft, das heißt: das Basic-Programm darf keine »unmöglichen« Koordinaten übermitteln.
— Der Bildschirm wird nur so weit gelöscht, wie es nötig ist.
— Die Zeropage-Addressierung wird in breitem Umfang benutzt.
— Der IRQ wird abgestellt.

Das wichtigste Mittel zur Erzeugung eines flüssigen Bilderablaufs und die Grundidee der Routine ist jedoch das »verdeckte Zeichnen«. Dafür werden zwei Bitmaps benötigt (bei unserem Programm ab $A000 und $E000). Während nun eine der beiden zu sehen ist, wird auf der anderen gezeichnet. Nun wird das neue Bild sichtbar gemacht und das alte, welches nicht mehr zu sehen ist, wird gelöscht. Dieser Vorgang wiederholt sich, bis alle Bilder abgearbeitet sind. Hier noch eine Anmerkung: die Line-Routine haben wir mit einigen Änderungen dem Artikel »Ein schneller Drawline-Algorithmus« aus dem 64'er, 4/84 entnommen. Die Funktion läßt sich am besten dort nachvollziehen. (Armin und Dirk Biernaczyk/tr)

Felder:	
X(I), Y(I), Z(I)	: Originalkoordinaten der Figur aus den DATA-Zeilen
X1(I), Y1(I), Z(I)	: Koordinaten beim Rechnen und für das POKEN
P1(I), P2(I)	: Verbindungsvorschriften aus den DATA-Zeilen
Normale Variablen:	
PO	: Speicherpointer für fertige Daten
Q-1	: Zähler für fertige Bilder
A2	: Anzahl der Linien pro Bild
A1	: Anzahl der Punkte pro Bild
AN	: Anzahl der Bilder
P1, P2, P3, P4, PB, PA	: Variablen zur Berechnung des zu löschenden Bildschirmbereichs
WX, WY, WZ, XA, YA, ZA, XB, YB, ZB	: Variablen für die Drehbewegungen und Verschieben (werden bei jedem Bild auf W1, ... aufaddiert)
W1, W2, W3, X1, Y1, Z1, X2, Y2, Z2	: Momentane Lage des Körpers

Variablenliste

```
100 REM -------------------------------     <087>
105 REM -- 3D-MOVIE-MAKER               --  <194>
110 REM -- EIN PROGRAMM VON:            --  <046>
115 REM -- DIRK & ARMIN BIERNACZYK --       <115>
120 REM -- AN DER PAPENBURG 41          --  <029>
125 REM -- 4630 BOCHUM 6                --  <208>
130 REM -- TEL.: //////                 --  <147>
135 REM -------------------------------     <122>
140 REM (C) 1985 BY ARMIN & DIRK            <153>
145 REM             BIERNACZYK              <134>
150 :                                       <126>
160 REM ------------------                  <017>
170 REM --- HAUPTMENUE ---                  <121>
180 REM ------------------                  <037>
190 :                                       <166>
191 IF A=0 THEN A=1:LOAD"TRICK.OBJ",8,1     <151>
192 :                                       <168>
195 POKE 56,50:CLR:REM SPEICHER HERAB.      <154>
200 PRINT CHR$(147)                         <229>
210 PRINT SPC(10)"** 3D-MOVIE-MAKER **"     <180>
220 PRINT:PRINT:PRINT:PRINT                 <015>
230 PRINT SPC(9)"1 - GRAFIK ERSTELLEN"      <194>
235 PRINT                                   <081>
240 PRINT SPC(9)"2 - GRAFIK ABSPIELEN"      <219>
245 PRINT                                   <091>
250 PRINT SPC(9)"3 - GRAFIK LADEN"          <115>
255 PRINT                                   <101>
260 PRINT SPC(9)"4 - GRAFIK ABSPEICHERN"    <129>
262 PRINT                                   <110>
264 PRINT SPC(9)"5 - ENDE"                  <041>
265 :                                       <243>
270 GET W$:IF W$<"1"OR W$>"5"THEN 270       <160>
280 W=VAL(W$)                               <125>
285 IF W=5 THEN END                         <001>
290 ON W GOSUB 1040,5040,6040,7040          <235>
300 GOTO 200                                <238>
310 :                                       <032>
315 :                                       <037>
320 :                                       <042>
1000 REM ------------------                 <141>
1010 REM --- ERSTELLEN ---                  <030>
1020 REM ------------------                 <161>
1030 :                                      <246>
1040 GOSUB 4540   :REM VARIABLEN            <022>
1050 GOSUB 4040   :REM EINLESEN             <139>
1060 PRINT CHR$(147)"FERTIGE BILDER: "      <131>
1061 IF AN*(A2*6+2)<23000 THEN 1080         <089>
1062 PRINT"ZU WENIG SPEICHERPLATZ"          <143>
1063 POKE 198,0:WAIT 198,1:RETURN           <190>
1070 :                                      <030>
1080 FOR Q=1 TO AN  :REM ANZAHL DER BILDER  <125>
1085 GOSUB 3040   :REM BEWEGEN              <092>
1090 GOSUB 2040   :REM BERECHNEN            <083>
1095 GET TA$:IF TA$=" "THEN IF Q>1 THEN SY
     S 49152,Q-1,A2,1                       <039>
1100 NEXT                                   <094>
1110 Q=Q-1                                  <014>
1120 RETURN                                 <162>
1130 :                                      <090>
1135 :                                      <095>
2000 REM ------------------                 <125>
2010 REM --- BERECHNEN ---                  <159>
2020 REM ------------------                 <145>
2030 :                                      <228>
2040 REM --- VERSCHIEBEN1 ---               <175>
2050 :                                      <250>
2060 FOR I=1 TO A1                          <183>
2070 X1(I)=X(I)+X1:Y1(I)=Y(I)+Y1            <171>
2080 Z1(I)=Z(I)+Z1                          <209>
2090 NEXT                                   <068>
2100 :                                      <044>
2110 REM --- DREHEN ---                     <196>
2120 :                                      <064>
2130 IF W1=0 THEN 2250                      <125>
2140 FOR I=1 TO A1                          <007>
2150 XD=X1(I):YD=Y1(I)                      <194>
2160 IF XD=0 THEN XD=1E-20                  <137>
2165 IF YD=0 THEN YD=1E-20                  <151>
2170 R=SQR(XD*XD+YD*YD)                     <156>
2180 W=ATN(YD/XD)                           <214>
2190 IF XD>0 AND YD<0 THEN W=W+π*2:GOTO 22
     10                                     <204>
2200 IF XD<0 THEN W=W+π                     <158>
2210 W=W+W1                                 <219>
2220 Y1(I)=SIN(W)*R:X1(I)=COS(W)*R          <236>
2230 NEXT                                   <208>
2240 :                                      <184>
2250 IF W2=0 THEN 2370                      <015>
2260 FOR I=1 TO A1                          <127>
2270 ZD=Z1(I):YD=Y1(I)                      <130>
2280 IF ZD=0 THEN ZD=1E-20                  <019>
2285 IF YD=0 THEN YD=1E-20                  <015>
2290 R=SQR(ZD*ZD+YD*YD)                     <152>
2300 W=ATN(YD/ZD)                           <082>
2310 IF ZD>0 AND YD<0 THEN W=W+π*2:GOTO 23
     30                                     <216>
2320 IF ZD<0 THEN W=W+π                     <040>
2330 W=W+W2                                 <214>
2340 Y1(I)=SIN(W)*R:Z1(I)=COS(W)*R          <103>
```

Listing 2. »3D-Movie-Maker«

```
2350 NEXT                                   <074>
2360 :                                      <050>
2370 IF W3=0 THEN 2510                      <133>
2380 FOR I=1 TO A1                          <249>
2390 ZD=Z1(I):XD=X1(I)                      <180>
2400 IF ZD=0 THEN ZD=1E-20                  <141>
2405 IF XD=0 THEN XD=1E-20                  <128>
2410 R=SQR(ZD*ZD+XD*XD)                     <130>
2420 W=ATN(XD/ZD)                           <140>
2430 IF ZD>0 AND XD<0 THEN W=W+π*2:GOTO 24
     50                                     <210>
2440 IF ZD<0 THEN W=W+π                     <160>
2450 W=W+W3                                 <206>
2460 X1(I)=SIN(W)*R:Z1(I)=COS(W)*R          <219>
2470 NEXT                                   <194>
2480 :                                      <170>
2490 REM --- UMRECHNEN ---                  <076>
2500 :                                      <190>
2510 FOR I=1 TO A1                          <123>
2530 X1(I)=(X1(I)+X2)*1.01↑(Z1(I)+Z2)       <215>
2540 Y1(I)=(Y1(I)+Y2)*1.01↑(Z1(I)+Z2)       <103>
2560 NEXT                                   <030>
2565 REM 1.01 KANN LEICHT GEAENDERT         <109>
2567 REM WERDEN                             <178>
2580 :                                      <016>
2590 PA=0:PB=199                            <043>
2600 PO=PO-2                                <006>
2601 REM -----------------------------      <048>
2602 REM --- UBERGETRETENE LINIEN  ---      <122>
2603 REM --- BERECHNEN UND POKEN   ---      <032>
2604 REM -----------------------------      <051>
2610 FOR I=1 TO A2                          <226>
2620 X0%=X1(P1(I)):Y1%=Y1(P1(I))            <199>
2630 X2%=X1(P2(I)):Y2%=Y1(P2(I))            <232>
2640 X1%=0:X3%=0:ME=0                       <087>
2641 IF X0%> 159 AND X2%> 159 THEN 2760     <013>
2642 IF X0%<-159 AND X2%<-159 THEN 2760     <041>
2643 IF Y1%<- 99 AND Y2%<- 99 THEN 2760     <177>
2644 IF Y1%>  99 AND Y2%>  99 THEN 2760     <186>
2650 IF Y1%=Y2%THEN 2711                    <005>
2655 IF X2%=X0%THEN 2690                    <065>
2660 M=(Y2%-Y1%)/(X2%-X0%)                  <177>
2670 B=-M*X0%+Y1%                           <127>
2680 GOTO 2720                              <008>
2690 IF Y1%>99 OR Y1%<-99 THEN Y1%=99*SGN(
     Y1%)                                   <171>
2700 IF Y2%>99 OR Y2%<-99 THEN Y2%=99*SGN(
     Y2%)                                   <229>
2710 GOTO 2760                              <166>
2711 IF X0%>159 OR X0%<-159 THEN X0%=159*S
     GN(X0%)                                <155>
2712 IF X2%>159 OR X2%<-159 THEN X2%=159*S
     GN(X2%)                                <029>
2713 GOTO 2760                              <169>
2720 IF X0%>159 OR X0%<-159 THEN X0%=159*S
     GN(X0%):Y1%=M*X0%+B                    <150>
2730 IF X2%>159 OR X2%<-159 THEN X2%=159*S
     GN(X2%):Y2%=M*X2%+B                    <057>
2740 IF Y1%>99 OR Y1%<-99 THEN Y1%=99*SGN(
     Y1%):X0%=(Y1%-B)/M                     <122>
2750 IF Y2%>99 OR Y2%<-99 THEN Y2%=99*SGN(
     Y2%):X2%=(Y2%-B)/M                     <248>
2760 IF Y1%>99 OR Y1%<-99 THEN X1%=255:X0%
     =0:Y1%=0:Y2%=0:X2%=0:GOTO 2810         <058>
2770 IF X0%>159 OR X0%<-159 THEN X1%=255:X
     0%=0:Y1%=0:Y2%=0:X2%=0:GOTO 2810       <028>
2780 :                                      <216>
2790 X0%=X0%+160:X2%=X2%+160                <212>
2791 Y1%=Y1%+100:Y2%=Y2%+100                <178>
2793 IF Y1%>PA THEN PA=Y1%                  <079>
2794 IF Y2%>PA THEN PA=Y2%                  <160>
2795 IF Y1%<PB THEN PB=Y1%                  <218>
2796 IF. Y2%<PB THEN PB=Y2%                 <043>
2799 IF X0%>255 THEN X0%=X0%-256:X1%=1      <247>
2800 IF X2%>255 THEN X2%=X2%-256:X3%=1      <060>
2805 :                                      <241>
2810 POKE PO,X0%:POKE PO-1,X1%              <039>
2820 POKE PO-2,Y1%:POKE PO-3,X2%            <031>
2830 POKE PO-4,X3%:POKE PO-5,Y2%            <179>
2840 PO=PO-6                                <252>
2850 NEXT                                   <066>
2860 :                                      <042>
2862 IF PA<PB THEN PA=199:PB=0              <088>
2863 PA=40*((PA OR 7)+1)/256+1              <027>
2864 PB=40*(PB AND 248)/256                 <219>
2866 POKE PO+6*A2+2,INT(P1)-INT(P3)         <168>
2868 POKE PO+6*A2+1,P3                      <198>
2870 P1=P2:P2=PA:P3=P4:P4=PB                <144>
2872 PRINT CHR$(19)SPC(16)Q                 <186>
2880 RETURN                                 <144>
2890 :                                      <072>
2900 :                                      <082>
3000 REM ---------------                    <080>
3010 REM --- BEWEGEN ---                    <186>
3020 REM ---------------                    <100>
3030 :                                      <212>
3040 IF E=0 THEN 3100                       <082>
3050 W1=W1+WZ:W2=W2+WX:W3=W3+WY             <027>
3060 X1=X1+XA:Y1=Y1+YA:Z1=Z1+ZA             <125>
3070 X2=X2+XB:Y2=Y2+YB:Z2=Z2+ZB             <143>
3080 E=E-1:RETURN                           <138>
3090 :                                      <018>
3100 READ XA,YA,ZA,WX,WY,WZ,XB,YB,ZB,E      <203>
3110 WX=WX*π/180:WY=WY*π/180                <254>
3120 WZ=WZ*π/180:YA=-YA:YB=-YB              <044>
3130 GOTO 3050                              <198>
3140 :                                      <068>
3150 :                                      <078>
4000 REM ----------------                   <244>
4010 REM --- EINLESEN ---                   <155>
4020 REM ----------------                   <008>
4030 :                                      <196>
4040 I=0                                    <163>
4050 I=I+1                                  <223>
4060 READ X(I),Y(I),Z(I)                    <224>
4070 Y(I)=-Y(I)                             <251>
4080 IF X(I)<1000 THEN 4050                 <116>
4090 A1=I-1                                 <124>
4100 I=0                                    <225>
4110 I=I+1                                  <029>
4120 READ P1(I),P2(I)                       <100>
4130 IF P1(I)<1000 THEN 4110                <115>
4140 A2=I-1                                 <184>
4150 :                                      <062>
4155 AN=0                                   <003>
4157 READ A:IF A=1000 THEN 4180             <172>
4160 FOR I=1 TO 9:READ A:NEXT               <013>
4170 AN=AN+A:GOTO 4157                      <030>
4180 RESTORE                                <166>
4190 READ A,A,A                             <059>
4200 IF A<1000 THEN 4190                    <182>
4210 READ A,A                               <164>
4220 IF A<1000 THEN 4210                    <082>
4230 :                                      <142>
4240 PRINT CHR$(147)"BILDERZAHL{2SPACE}"AN  <102>
4250 PRINT CHR$(19)SPC(11);:INPUT I         <252>
4260 IF I>255 OR I>AN THEN 4240             <040>
4270 AN=I                                   <153>
4280 RETURN                                 <018>
4290 :                                      <202>
4300 :                                      <212>
4500 REM -----------------                  <085>
4510 REM --- VARIABLEN ---                  <042>
4520 REM -----------------                  <105>
4530 :                                      <188>
4533 REM NACH BEDARF DIMENSIONIEREN         <188>
4536 :                                      <194>
4540 DIM X(50),Y(50),Z(50)                  <164>
4560 DIM X1(50),Y1(50),Z1(50)               <010>
4570 DIM P1(50),P2(50)                      <048>
4580 PO=35839:P1=32:P2=32:P3=0:P4=0         <246>
4590 RETURN                                 <074>
4600 :                                      <002>
4605 :                                      <007>
5000 REM -----------------                  <077>
5010 REM --- ABSPIELEN ---                  <165>
5020 REM -----------------                  <097>
5030 :                                      <180>
5040 IF Q>0 AND A2>0 THEN 5090              <139>
5050 PRINT CHR$(147):PRINT:PRINT:PRINT      <238>
5060 PRINT SPC(5)"ES GIBT KEINE GRAFIK"     <094>
5070 POKE 198,0:WAIT 198,1:POKE 198,0       <091>
5080 RETURN                                 <056>
5090 INPUT"{CLR}WIEVILE DURCHLAUEFE";DU     <209>
5091 IF DU>255 OR DU<1 THEN 5090            <161>
5092 PRINT"{HOME,2DOWN}WIEVIELE BILDER{5SP
     ACE}"Q                                 <097>
5093 PRINT"{UP}"SPC(19);:INPUT I            <233>
5094 IF I>Q OR I<1 THEN 5092                <171>
5100 :                                      <250>
```

Listing 2. Das Basic-Programm zum »3D-Movie-Maker«. Verwenden Sie zur Eingabe bitte den Checksummer.

```
5105 SYS 49152,I,A2,DU:RETURN                <131>
5110 :                                       <004>
6000 REM -------------                       <153>
6010 REM --- LADEN ---                       <096>
6020 REM -------------                       <173>
6030 :                                       <164>
6040 PRINT CHR$(147):PRINT:PRINT:PRINT:PRI
     NT                                      <138>
6050 INPUT"{3SPACE}FILENAME: ";NA$           <205>
6055 IF NA$="M"THEN RETURN                   <109>
6060 OPEN 2,8,2,NA$+".GRA,S,R"               <146>
6070 OPEN 1,8,15:INPUT#1,FE$                 <136>
6080 IF FE$="00"THEN 6090                    <007>
6082 CLOSE 1:CLOSE 2:GOTO 6040               <180>
6090 GET#2,Q$,A2$                            <060>
6100 Q=ASC(Q$):A2=ASC(A2$)                   <177>
6110 AD=35839-Q*(A2*6+2)                     <039>
6140 CLOSE 2:CLOSE 1                         <064>
6150 AH=INT(AD/256):AL=AD-AH*256             <100>
6160 A$=NA$+".GRA,S"                         <191>
6170 FOR I=51000 TO 51000+LEN(A$)-1          <155>
6180 POKE I,ASC(MID$(A$,I-50999,1))          <128>
6190 NEXT                                    <104>
6200 POKE 183,LEN(A$)                        <167>
6210 POKE 187,56:POKE 188,199                <208>
6220 POKE 185,0:POKE 186,8:POKE 147,0        <250>
6570 POKE 195,AL:POKE 196,AH                 <154>
6580 SYS 62648                               <035>
6590 RETURN                                  <042>
6600 :                                       <226>
6610 :                                       <236>
6620 :                                       <246>
7000 REM ------------------                  <164>
7010 REM --- ABSPEICHERN ---                 <245>
7020 REM ------------------                  <184>
7030 :                                       <148>
7040 IF Q>0 AND A2>0 THEN 7090               <139>
7050 PRINT CHR$(147):PRINT:PRINT:PRINT       <206>
7060 PRINT SPC(5)"ES GIBT KEINE GRAFIK"      <062>
7070 POKE 198,0:WAIT 198,1:POKE 198,0        <059>
7080 RETURN                                  <024>
7090 PRINT CHR$(147):PRINT:PRINT:PRINT       <246>
7100 INPUT"{4SPACE}FILENAME: ";NA$           <239>
7105 IF NA$="M"THEN RETURN                   <143>
7110 OPEN 2,8,2,NA$+".GRA,S,W"               <084>
7120 OPEN 1,8,15:INPUT#1,FE$                 <170>
7130 IF FE$="00"THEN 7140                    <024>
7135 CLOSE 1:CLOSE 2:GOTO 7090               <003>
7140 PRINT#2,CHR$(Q);CHR$(A2);               <185>
7150 FOR I=35839-Q*(A2*6+2)TO 35839          <228>
7160 PRINT#2,CHR$(PEEK(I));:NEXT             <135>
7170 CLOSE 2:CLOSE 1:RETURN                  <251>
7175 :                                       <039>
7180 :                                       <044>
7185 :                                       <049>
8000 REM -------------------------           <025>
8010 REM --- DATAS FUER PUNKTE ---           <255>
8020 REM -------------------------           <045>
8030 :                                       <132>
8040 DATA -20, 10,0                          <049>
8050 DATA -30, 10,0                          <075>
8060 DATA -30,-10,0                          <205>
8070 DATA -20,-10,0                          <199>
8080 DATA -20,  0,0                          <029>
8090 DATA -30,  0,0                          <055>
8100 DATA -15, 10,0                          <253>
8110 DATA -15,  0,0                          <203>
8120 DATA - 5, 10,0                          <053>
8130 DATA - 5,  0,0                          <161>
8140 DATA - 5,-10,0                          <132>
8150 DATA  10, 10,0                          <035>
8160 DATA   5,  5,0                          <096>
8170 DATA  15,-10,0                          <194>
8180 DATA   5,-10,0                          <064>
8190 DATA   5,  0,0                          <222>
8200 DATA  15,  0,0                          <103>
8210 DATA  15, -5,0                          <128>
8220 DATA   5, -5,0                          <243>
8230 DATA  20,-10,0                          <205>
8240 DATA  20,  0,0                          <107>
8250 DATA  20, -5,0                          <132>
8260 DATA  30,  0,0                          <131>
8900 DATA 1000,1000,1000                     <216>
8910 :                                       <250>
9000 REM ----------------------------        <203>
9010 REM --- VERBINDUNGSVOSCHRIFT ---        <134>
9020 REM ----------------------------        <223>
9030 :                                       <116>
9040 DATA  1, 2,  2, 3,  3, 4,  4, 5         <055>
9050 DATA  5, 6,  7, 8,  8,10,  9,11         <122>
9060 DATA 12,13, 14,15, 15,16, 16,17         <021>
9070 DATA 17,18, 18,19, 20,21, 22,23         <243>
9450 DATA 1000,1000                          <198>
9460 :                                       <036>
9500 REM -------------------------           <211>
9510 REM --- BEWEGUNGSVOSCHRIFT ---          <187>
9520 REM -------------------------           <231>
9530 :                                       <108>
9531 REM XA,YA,ZA,WX,WY,WZ,XB,YB,ZB,E        <060>
9550 DATA 0,0,  1 ,0,0,0 ,0,0,  0 ,30        <043>
9560 DATA 0,0,  0 ,0,5,0 ,0,0,  0 ,108       <185>
9570 DATA 0,0,-30 ,0,0,0 ,0,0,-30 ,1         <233>
9580 DATA 0,0,  0 ,5,0,5 ,0,0,.81 ,36        <178>
9620 DATA 1000                               <138>
```


Listing 2. »3D-Movie-Maker« (Schluß)

```
 programm : trick.obj     c000 c2a4
-------------------------------------

c000 : 20 36 c0 20 a4 c0 20 98  71
c008 : c0 20 e6 c0 c6 4f d0 f3  bc
c010 : 20 7f c0 c6 52 d0 ec a9  ab
c018 : 37 85 01 58 a9 00 85 c6  9b
c020 : a5 c6 f0 fc c6 c6 a9 1b  83
c028 : 8d 11 d0 a9 15 8d 18 d0  67
c030 : a9 03 8d 00 dd 60 20 fd  1c
c038 : ae 20 9e b7 86 4e 20 fd  ec
c040 : ae 20 9e b7 86 50 20 fd  04
c048 : ae 20 9e b7 86 52 a9 3b  bd
c050 : 8d 11 d0 a9 38 8d 18 d0  c1
c058 : a9 00 8d 00 dd 20 c2 c0  d0
c060 : 78 a9 35 85 01 a2 0b bd  78
c068 : 8c c0 95 a0 ca 10 f8 a2  24
c070 : 20 a9 00 a8 20 a0 00 a9  d4
c078 : a0 85 fc a9 00 85 53 a9  1c
c080 : 8b 85 8e a9 fa 85 8d a5  04
c088 : 4e 85 4f 60 99 00 e0 c8  27
c090 : d0 fa e6 a2 ca d0 f5 60  b7
c098 : a5 50 85 51 20 70 c1 c6  0b
c0a0 : 51 d0 f9 60 a0 05 b1 8d  f8
c0a8 : aa 88 b1 8d 05 fc 85 a2  48
c0b0 : 38 a5 8d e9 02 85 8d b0  3f
c0b8 : 02 c6 8e a9 00 a8 20 a0  fd
c0c0 : 00 60 a9 10 a0 cc 84 8c  f8
c0c8 : a0 00 84 8b 20 d9 c0 a0  10
c0d0 : 8c 84 8c a0 00 20 d9 c0  bf
c0d8 : 60 a2 04 91 8b c8 d0 fb  f7
c0e0 : e6 8c ca d0 f6 60 a5 53  89
c0e8 : d0 0c e6 53 a9 e0 85 fc  94
c0f0 : a5 53 8d 00 dd 60 c6 53  45
c0f8 : a9 a0 4c ee c0 8a 4a 4a  00
c100 : 29 fe a8 b9 36 c1 85 fd  8d
c108 : b9 37 c1 85 fe 8a 29 07  75
c110 : 18 65 fd 85 fd a5 14 29  ba
c118 : f8 65 fd 85 fd a5 fe 65  c7
c120 : fc 65 15 85 fe a5 14 29  85
c128 : 07 49 07 aa bd 68 c1 a0  52
c130 : 00 11 fd 91 fd 60 00 00  4d
c138 : 40 01 80 02 c0 03 00 05  87
c140 : 40 06 80 07 c0 08 00 0a  e5
c148 : 40 0b 80 0c c0 0d 00 0f  42
c150 : 40 10 80 11 c0 12 00 14  9f
c158 : 40 15 80 16 c0 17 00 19  fd
c160 : 40 1a 80 1b c0 1c 00 1e  5a
c168 : 01 02 04 08 10 20 40 80  71
c170 : 20 71 c2 a0 01 84 62 84  d4
c178 : 5f 84 5d 88 84 5e 84 61  92
c180 : 84 60 88 a5 59 c5 15 90  44
c188 : 08 d0 18 a5 58 c5 14 b0  18
c190 : 12 38 a5 14 e5 58 85 5b  98
c198 : a5 15 e5 59 85 5c 84 62  7e
c1a0 : 4c b0 c1 38 a5 58 e5 14  99
c1a8 : 85 5b a5 59 e5 15 85 5c  45
c1b0 : a5 ff c5 57 b0 0c 38 a5  49
c1b8 : 57 e5 ff 85 5a 84 5f 4c  92
c1c0 : c6 c1 e5 57 85 5a a5 5c  46
c1c8 : d0 19 a5 5b c5 5a b0 13  12
c1d0 : a6 5a 85 5a 86 5b a5 62  ef
c1d8 : 85 60 a5 5f 85 61 c8 84  72
c1e0 : 62 84 5f a5 5c 4a 85 59  f2
c1e8 : a5 5b 6a 85 58 4c 5c c2  65
c1f0 : a5 62 30 0b 18 65 14 85  3c
c1f8 : 14 90 0d e6 15 d0 09 18  a1
c200 : 65 14 85 14 b0 02 c6 15  b4
c208 : 18 a5 57 65 61 85 57 18  45
c210 : a5 58 65 5a 85 58 a5 59  ea
c218 : 69 00 85 59 e6 5d d0 02  ae
c220 : e6 5e a5 59 c5 5c 90 34  b4
c228 : d0 06 a5 5b c5 58 b0 2c  0a
c230 : 38 a5 58 e5 5b 85 58 a5  9c
c238 : 59 e5 5c 85 59 a5 60 30  f0
c240 : 0b 18 65 14 85 14 90 0d  89
c248 : e6 15 d0 09 18 65 14 85  16
c250 : 14 b0 02 c6 15 18 a5 57  6d
c258 : 65 5f 85 57 a6 57 20 fd  5b
c260 : c0 a5 5e c5 5c 90 07 a5  f5
c268 : 5b c5 5d b0 01 60 4c f0  39
c270 : c1 a0 05 b1 8d 85 14 88  5f
c278 : b1 8d 10 05 68 68 4c 97  bf
c280 : c2 85 15 88 b1 8d 85 57  a7
c288 : 88 b1 8d 85 58 88 b1 8d  a8
c290 : 85 59 88 b1 8d 85 ff 38  90
c298 : a5 8d e9 06 85 8d b0 02  ca
c2a0 : c6 8e 60 20 0f c9 0e d0  e3
```

Listing 1. Die Maschinenroutine zu »3D-Movie-Maker«. Bitte mit dem MSE eingeben.

Verbotene Variablen

Hatten Sie schon Probleme mit dem SYNTAX ERROR in anscheinend korrekten Basic-Zeilen? Wahrscheinlich lag es an verbotenen Variablen, die hier näher erklärt werden. Außerdem zeigen wir Ihnen, wie Sie Basic-Programme strukturieren können.

Wer jemals versucht hat, fremde Basic-Programme zu verstehen und dem Gedankengang des Autors zu folgen, der hat sich bestimmt des öfteren darüber geärgert, daß Basic leider allzuoft in einem Spaghetticode ausartet. Sogar der Autor eines Programms findet sich, wenn er nicht 'zig REM-Zeilen eingearbeitet hat, nach einigen Monaten nicht mehr mit dem eigenen Programm zurecht. Das sind dann die Momente, wo man in Richtung Pascal oder all den anderen strukturierten Programmiersprachen schielt.

Der vorliegende Artikel wendet sich an alle, die nach einer Möglichkeit suchen, mit dem vorhandenen (zugegebenermaßen mageren) Basic des C 64 trotzdem ihre Listings etwas übersichtlicher zu gestalten und ihnen eine Struktur zu verleihen.

Übersichtliche Listings

Einige Basic-Programmierer benutzen zur Strukturierung ihrer Programme entweder REMs oder einfach nur den Doppelpunkt. Damit können einzelne Programmteile beim Listing optisch voneinander getrennt werden. Eine bessere Lesbarkeit wird erzielt. Eine andere Möglichkeit zur Strukturierung bieten die Grafikzeichen des C 64.

Nun, werden Sie sich fragen, was denn so besonderes daran sei, Grafikzeichen in Programmen zu verwenden. Man sieht sie doch alle Tage in Listings. Der Witz dieser Lösung zur optischen Aufbereitung von Listings ist, daß die verwendeten Grafikzeichen unsichtbar bleiben, aber die Lesbarkeit der Programme sichtbar unterstützen.

Um diesen scheinbaren Widerspruch aufzulösen, ist es notwendig, sich ein wenig mit der Art und Weise zu beschäftigen, wie der Basic-Interpreter Programmzeilen im Basic-Speicher ab $0801 (dezimal 2049) ablegt.

Schauen Sie sich dazu einmal das kleine Beispielprogramm an.

```
10  FOR I=0 TO 20
20  PRINT"TEST"
30  NEXT
```

Mit einem Monitor erhalten Sie den folgenden Speicherauszug:

```
0800  00  10  08  0A  00  81  20  49
0808  B2  31  20  A4  20  32  30  00
0810  1C  08  14  00  99  22  54  45
0818  53  54  22  00  22  08  1E  00
0820  82  00  00  00
```

Da das Beispielprogramm klein ist, wird noch keine Strukturierung sinnvoll. Stellen Sie sich aber vor, es wäre ein Programm mit 50 und mehr Zeilen, mit einer Menge von GOTOs, GOSUBs und FOR-NEXT-Schleifen. In diesem Fall wird ein herkömmliches Listing von Zeile zu Zeile unübersichtlicher.

Eine einfache, aber recht wirkungsvolle Möglichkeit, ein Listing »durchsichtig« zu machen, ist das Einrücken der Programmzeilen innerhalb einer Schleife. Das Einrücken geschieht meist durch die Voreinstellung eines oder mehrerer Doppelpunkte am Anfang einer Zeile in Verbindung mit einer Reihe von Leerzeichen.

Es geht aber auch anders. Geben Sie dazu die Zeile 20 neu ein:

20 {SHIFT-J} {4*Leertaste} PRINT"TEST"

Wenn Sie nun das kleine Programm listen, ist etwas Erstaunliches passiert:

```
10  FOR I=1 TO 20
20      PRINT"TEST"
30  NEXT
```

Das bei der Eingabe sichtbare Grafikzeichen ist unsichtbar geworden, der PRINT-Befehl aber ist wie bei der Eingabe um vier Leerzeichen (plus ein erzwungenes Leerzeichen nach der Zeilennummer) nach rechts gerückt worden. Wenn Sie das gleiche durch Eingabe von vier Leerschritten unmittelbar nach der Zeilennummer versucht hätten, so wäre der Versuch fehlgeschlagen.

Der Interpreter ignoriert nämlich alle Leerzeichen, die der Programmierer nach der Zeilennummer eingibt; vielmehr wirft er diese alle weg, bis er auf die erste Basic-Aweisung oder einem Doppelpunkt in der betreffenden Zeile trifft. Andererseits besteht er auf genau einem Leerzeichen unmittelbar nach der Zeilennummer, das er eventuell selbst einfügt.

Aber was ist durch das Grafikzeichen im Speicher passiert? Einen Speicherauszug liefert das folgende Bild:

```
0800  00  10  08  0A  00  81  20  49
0808  B2  31  20  A4  20  32  30  00
0810  20  08  14  00  20  20  20  20
0818  99  22  54  45  53  54  22  00
0820  26  08  1E  00  82  00  00  00
```

Die interessantesten Speicherstellen beginnen ab $0810, wo sich gegenüber unserem vorangehenden Beispiel etwas geändert hat. Wir sehen in $0810 und $0811 den Zeiger (hier 20 08 = $0820) auf den Anfang der nächsten Basic-Zeile. Die folgenden Hex-Zeichen 14 00 ergeben in Low-/High-Byte-Darstellung die Zeilennummer 20. Dann folgen die vier eingefügten Leerzeichen ($20) und erst danach das Token 99 (interne Abkürzung des Interpreters) für den PRINT-Befehl. Die folgenden Zeichen sind dann, jeweils um vier Speicherstellen verschoben, mit den Zeichen aus dem vorangehenden Beispiel identisch.

Wenn Sie nun spaßeshalber den Cursor nochmals auf die Zeile 20 des Programmes fahren und die RETURN-Taste drücken, schaut die Zeile 20 nach erneutem Listen wieder so aus, wie ohne Benutzung des »Grafikzeichen-Tricks«. Das geschieht, weil der C 64 einen bildschirmorientierten Editor besitzt, der in letzterem Fall nicht unterscheidet, ob die Zeile 20 neu eingetippt oder »vom Bildschirm übernommen« wird.

Mit dem Eingeben eines Grafikzeichens unmittelbar nach der Zeilennummer übertölpeln Sie den Interpreter, der diese Art von Zeichen an dieser Stelle nicht erwartet hätte. Vor lauter Staunen »vergißt« er dann die eingegebenen Leerzeichen zu ignorieren, wie es sonst seine Art ist.

Verwendung von »verbotenen« Variablen

Hand aufs Herz! Ist es Ihnen nicht auch schon einmal passiert, daß Sie ein Programm geschrieben haben und der Lesbarkeit halber leicht merkbare und selbsterklärende Variablennamen verwendeten. Beim ersten Testlauf stieg der C 64 dann mit der Fehlermeldung SYNTAX ERROR aus und Sie in die Fehlersuche ein.

Meist dauert es dann ein wenig, bis man den Fehler erkennt und beispielsweise einen Variablennamen wie SCHIFF als den Übeltäter entlarvt. SCHIFF ist ein verbotener Name für eine Variable, da der Name eine Basic-Anweisung (IF) enthält. Stößt der Interpreter auf einen solchen Namen, so ist er irritiert, meldet einen Fehler und beendet einfach seine Arbeit.

Schauen Sie sich anhand eines kleinen Beispiels einmal die Arbeitsweise des Interpreters an.

```
10  SCHIFF$="TITANIC"
20  PRINT SCHIFF$
```

Wenn Sie das Programm starten, so geht das Programm ebenso unter wie das besagte Schiff. Ein Monitorlisting schafft Klarheit:

```
0800  00  16  08  0A  00  53  43  48
0808  8B  46  24  B2  22  54  49  54
0810  41  4E  49  43  22  00  23  08
0818  14  00  99  20  53  43  48  8B
0820  46  24  00  00  00
```

Genauer die Speicherzelle $0808 (beziehungsweise $081F). Der C 64 übernimmt eine Programmzeile erst in seinen Programmspeicher, nachdem man die RETURN-Taste gedrückt hat. Dann wird vom Betriebssystem die eingegebene Zeile Zeichen für Zeichen vom Bildschirm übernommen. Der Interpreter hat also beim Eingeben des oben angeführten Programms die Zeichenfolge SCHIFF$ ASCII-Zeichen für Zeichen in die Zellen ab $0805 eingeschrieben, bis er auf die für ihn bekannte Zeichenfolge IF traf. Folgerichtig schrieb er für IF das Token $8B ein und übernahm den Rest der Zeile wie gewohnt. Beim Abarbeiten des Programms versucht der Interpreter, die im Programmspeicher vorgefundenen Zeichen wieder seriell zu lesen und die einzelnen Befehle abzuarbeiten.

Dies mißlingt aber bereits in der ersten Zeile, weil er die Folge (ASCII)SCH(BASIC)IF(ASCII)F$ nicht zu interpretieren vermag. Weil der Interpeter in diesen Fällen so unnachgiebig ist, sind also alle Variablennamen verboten, die Basic-Wörter enthalten (zum Beispiel OR oder TAN).

Beispiele für verbotene Namen sind also: **TAN**NE, **LAND**$, **OR**T$, **ST**ART, um nur einige zu nennen. Wie man an der kleinen Aufzählung erkennt, muß man ziemlich aufpassen, um nicht wieder vom »SYNTAX-TERROR« erwischt zu werden.

Die Lesbarkeit von Programmen leidet schon ein wenig unter dieser Einschränkung, da man sehr oft auf unverständlichere und weniger selbsterklärende Variablennamen ausweichen muß. Mit Hilfe eines kleinen Tricks läßt sich der Interpreter aber so »über's Ohr hauen«, daß er verbotene Variablennamen akzeptiert und dem Programmierer quasi freie Auswahl für seine Variablennamen gibt.

Wer aufmerksam den ersten Teil dieses Artikels gelesen hat, kann sich vielleicht schon denken, wie dieser Trick funktioniert. Richtig, es werden wieder die Grafikzeichen an den entsprechenden Stellen eingesetzt, um den Interpreter von unseren Absichten zu überzeugen.

Geben Sie also ein:

```
10  SCHI{SHIFT-J}F$="TITANIC"
20  PRINT SCHI{SHIFT-J}FF$
```

Ein RUN bestätigt, daß unser Schiff soeben auf den Namen TITANIC getauft wurde und die SYNTAX-ERROR-Meldung des C 64 ausbleibt. Der Monitor zeigt, was mit dem Programm, das sich beim Listen äußerlich in nichts vom vorstehenden Beispiel unterscheidet, geschehen ist.

```
0800  00  17  08  0A  00  53  43  48
0808  49  46  46  24  B2  22  54  49
0810  54  41  4E  49  43  22  00  25
0818  08  14  00  99  20  53  43  48
0820  49  46  46  24  00  00  00
```

Ein Blick auf die Speicherstellen ab $0808 und $0820 zeigt, daß der Interpreter sich überlisten ließ und SCHIFF$ als sieben ASCII-Zeichen gespeichert hat. So kann er das Programm auch ausführen, weil er nicht mehr durch das Basic-Token IF verwirrt wird.

Das Drücken der RETURN-Taste bewirkt, daß der Interpreter jedes einzelne Zeichen der aktuellen Programmzeile übernimmt und auf Zeichenfolgen achtet, die ihm als Basic-Anweisungen oder -Funktionen bekannt sind. Letztere würde er als Token abspeichern. Nachdem er die Zeichenfolge SCH gelesen hat, hat er bereits eine Menge Arbeit hinter sich gebracht.

Die Folge SC kennt er nicht, wohl aber die Folge CH. Als nächste Zeichen könnten R und $ dann CHR$ ergeben, was als Token $C7 abgespeichert würde. Aber da er ein I als nächstes Zeichen liest, sagt er sich, vergessen wir die vorangehende Zeichenfolge und legen diese als ASCII-Code im Speicher ab. Bei I allerdings hat der Interpreter wieder seine Ohren gespitzt, denn dies könnte der Anfang von IF (Token $8B) oder auch INPUT (Token $85), INPUT# (Token $84) oder INT (Token $B5) sein.

Der Trick besteht darin, an geeigneter Stelle innerhalb des verbotenen Variablennamens ein Grafikzeichen einzugeben, um die Zeichenfolge so aufzutrennen, daß der Interpreter kein Basic-Wort (Token) erkennen kann.

Bei der Benutzung dieses Tricks ist streng zu beachten, daß jeder Variablenname genau gleich eingegeben wird, damit die Abspeicherung überall gleich ist, wenn ein- und dieselbe Variable mehrmals in einem Programm benutzt wird. Natürlich gilt trotz Benutzung des Tricks mit dem Grafikzeichen weiterhin, daß der C 64 nur die ersten beiden Zeichen eines Variablennamens auswertet. Deshalb ist die Benutzung von TONNE und TORTE im gleichen Programm trotz Eingabe mit unserem Trick nicht eindeutig und kann zu Problemen führen. Weiterhin ist bei zeitkritischen Programmen die Verwendung einstelliger Variablennamen günstiger. Wie weit allerdings die Verwendung von verbotenen Variablen sinnvoll ist, bleibt Ihnen überlassen. War nun das »GOTO« ein Befehl oder eine Variable? Eine Frage, die sich häufiger stellen wird, beim sorglosen Umgang mit verbotenen Variablen. Denkbar ist auch, mit den »verbotenen Variablen« ein Listing nahezu unlesbar zu gestalten (IF GOTO=1 THEN GOTO THEN, etc.), obwohl es optisch wunderbar strukturiert aussieht. Zum Schluß noch eine Bitte der Redaktion. Schicken Sie uns keine derart manipulierten Listings. Es kennt schließlich nicht jeder den Trick mit den Grafikzeichen.

(Dipl.-Ing. Raimund Trierscheid/hm)



Exsort — Sortieren mit Komfort

Mit Exsort werden sowohl numerische als auch alphanumerische Felder auf- oder und absteigend sehr schnell sortiert. Ein weiterer Clou: Ein zweites Feld kann abhängig vom ersten Feld mitsortiert werden.

Jeder Programmierer ärgert sich irgendwann einmal über das langsame Basic, das vor allem beim Suchen und Sortieren stört. Gute Sortierroutinen, in Assembler geschrieben, kann nicht jeder entwickeln. Viele Sort-Programme sind aber auch sehr einseitig: Entweder sortieren sie nur aufsteigend oder lediglich alphanumerische Felder. Exsort kann beides und noch mehr.

Vorteile:

- Zirka zehnmal so schnell wie die schnellste Basic-Version.
- Die Befehle können in jedem Basic-Programm angewendet werden.
- Unterprogramme in Basic, die oft nur ein bestimmtes Feld in einer Richtung sortieren können, entfallen.
- Die Erweiterung belegt keinen Basic-Speicher.
- Beim Sortieren von Strings kommt es nicht zu einer zeitraubenden Garbage-Collection, da die Descriptoren vertauscht werden.
- Ein zweites Feld, das Informationen über das erste Feld enthält, kann mitsortiert werden.
- Das zu sortierende Feld kann numerisch oder alphanumerisch sein.

Nachteile:

— Es kann nicht mit Exbasic oder Simons Basic zusammen genutzt werden.
— Es kann nicht compiliert werden.

Sortierbefehle

1. Befehl »so«

Syntax: so, (feldname), (anfangsindex), (endindex), (sortierungsrichtung)

Dieser Befehl sortiert ein beliebiges eindimensionales Feld innerhalb von zwei Grenzen mit einer vom Benutzer gewählten Sortierungsrichtung.

Beispiel 1:

Das Feld heißt ax$, alphanumerisch aufsteigend sortieren (von Index 100 bis Index 5000).

Befehl: so,ax$,100,5000,1 (1=aufsteigend)

Beispiel 2:

Das Feld heißt qe%, numerisch absteigend sortieren (von Index 0 bis zu dem Index, der in der Variable »en« enthalten ist).

Befehl: so,qe%,0,en,0 (0=absteigend)

Option: Manchmal ist es notwendig, daß Daten, die in einem zweiten Feld vorhanden sind, entsprechend dem ersten Feld sortiert werden.

Syntax: so,(feldname 1), (anfangsindex), (endindex), (sortierungsrichtung), (feldname2)

Beispiel: Das Feld fe$ soll alphanumerisch aufsteigend von Index 0 bis Index 10 sortiert werden. Die Daten in dem Realfeld »nr« sollen entsprechend dem ersten Feld sortiert werden.

Befehl: so,fe$,0,10,1,nr

2. Befehl: »se«

Syntax: se,(feldname),(anfangsindex),(endindex),(element)

Dieser Befehl durchsucht ein beliebiges eindimensionales Feld innerhalb von zwei Grenzen nach einem Element.

Beispiel: Es soll die Zahl - 12 in dem Feld rt% von Index 0 bis Index 100 gesucht werden.

Befehl: se,rt%,0,100, - 12

Wenn das Element gefunden wird, enthält die Variable »in« den jeweiligen Index. Wird das Element nicht gefunden, so enthält »in« den Wert —1.

Fehlermeldungen:

- type mismatch: Sie versuchten, einen String in einem numerischen Feld zu suchen (oder umgekehrt).
- wrong index: Beim Suchen war der Anfangsindex größer als der Endindex.
- bad subscript: Index außerhalb des zulässigen Bereiches.
- only one dimension array: Sie können nur eindimensionale Felder durchsuchen oder sortieren.
- array not found: Das Feld war nicht durch einen DIM-Befehl dimensioniert worden.
- wrong array name: Geben Sie bitte nur die ersten beiden Buchstaben des Feldnamen ein (plus % oder $ wenn nötig). Es wird dann sicher funktionieren.
- wrong sorting direction error: Sie haben einen anderen Wert als 0 oder 1 als Sortierungsrichtung angegeben.

Zu den Programmen:

Listing 1

Das Programm »Exsort« geben Sie bitte mit dem MSE ein. Beachten Sie dazu die Eingabegehinweise auf Seite 18.

Listing 2

Das Programm »Exsort demo« lädt das Absolutprogramm »Exsort« nach und startet es. Danach folgt eine Demonstration der beiden Befehle.

Um »Exsort« zu laden, muß Zeile 0 des Basic-Programms lauten:

0 if k=0 then k=1 : load"exsort.",8,1 (für Diskette)
0 if k=0 then k=1 : load"exsort.",1,1 (für Kassette)

In Zeile 1 muß stehen:

1 sys 49400

Da die Erweiterung nur einmal geladen und gestartet werden muß, kann sie bei späteren Starts des Programms übersprungen werden.

(Marcus Rickert/gk)

```
0 IF K=0 THEN K=1:LOAD"EXSORT?",8,1:REM LA
  DEN VON EXSORT                                <095>
1 SYS 49400:REM STARTEN VON EXSORT              <051>
100 REM ************************                <241>
110 REM ***    EXSORT DEMO    ***               <073>
120 REM ************************                <005>
130 REM                                         <192>
140 REM ******************                      <183>
150 REM * 1.BEFEHL: "SO" *                      <090>
160 REM ******************                      <203>
165 PRINT"{CLR}ERSTER BEFEHL: 'SO'"             <110>
170 INPUT"{DOWN}ZAHL DER ZU SORTIERENDEN E
    LEMENTE";A                                  <227>
175 PRINT"{DOWN}ANFUELLEN DES FELDES 'ZA'
    MIT ZUFALLS-{4SPACE}ZAHLEN"                 <101>
180 DIM ZA(A)                                   <157>
190 REM *** DAS FELD ZA WIRD MIT ZUFALLSZA
    HLEN BELEGT ***                             <109>
200 FOR S=1 TO A                                <130>
210 :ZA(S)=RND(1)*10000-5000                    <020>
220 NEXT S                                      <128>
230 IT=TI:REM ZEIT SPEICHERN                    <224>
240 PRINT"{DOWN}SORTIERBEGINN"                  <046>
250 REM                                         <056>
260 REM *** AUFRUF DES BEFEHLS "SO" ***         <125>
270 SO,ZA,1,A,1                                 <076>
```

```
280 REM  SO    =   BEFEHL                            <018>
290 REM  ZA    =   FELDNAME                          <181>
300 REM  1     =   ANFANGSINDEX                      <162>
310 REM  A     =   ENDINDEX                          <085>
320 REM  1     =   SORTIERUNGSRICHTUNG(AUF
    STEIGEND)                                        <098>
330 REM                                              <138>
340 IT=TI-IT                                         <234>
350 PRINT"{DOWN}SORTIERENDE"                         <221>
355 FOR S=1 TO 1000:NEXT S                           <202>
360 REM *** AUSGABE DER SORTIERTEN ELEMENT
    E ***                                            <068>
370 FOR S=1 TO A                                     <046>
380 :PRINT S,TAB(6)ZA(S)                             <108>
390 NEXT S                                           <044>
400 PRINT"{DOWN}ZEIT:"IT/60"SEC"                     <188>
410 PRINT"{DOWN}BITTE TASTE DRUECKEN"                <128>
420 GET T$:IF T$=""THEN 420                          <041>
430 REM                                              <238>
440 REM ******************                           <229>
450 REM * 2.BEFEHL: "SE" *                           <147>
460 REM ******************                           <249>
470 REM                                              <022>
480 CLR:DIM FE$(10000)                               <214>
485 PRINT"{CLR}ZWEITER BEFEHL 'SE'"                  <222>
490 REM ** IN 50 BELIEBIGE ELEMENTE   **             <162>
500 REM ** DES FELDES FE$ WIRD DAS    **             <137>
510 REM ** WORT "HALLO" GESCHRIEBEN   **             <253>
520 REM                                              <074>
525 PRINT"{DOWN}IN 50 BELIEBIGE ELEMENTE V
    ON FE$ WIRD{3SPACE}'HALLO' GESCHRIEBEN
    "                                                <128>
530 FOR S=1 TO 50                                    <250>
540 :FE$(RND(1)*10000)="HALLO"                       <158>
550 NEXT S                                           <206>
560 PRINT"{DOWN}IN FOLGENDEM ELEMENTEN STE
    HT 'HALLO':"                                     <155>
570 REM                                              <124>
580 REM ** AUSDRUCKEN JEDES INDEXES **               <184>
590 REM ** IN DEM "HALLO" STEHT     **               <081>
600 REM                                              <154>
610 IN=-1:IT=TI                                      <247>
620 REM ** AUFRUF DES BEFEHLS "SE" **                <200>
630 SE,FE$,IN+1,10000,"HALLO"                        <234>
640 REM  SE       =   BEFEHL                         <220>
650 REM  FE$      =   FELDNAME                       <182>
660 REM  IN+1     =   ANFANGSINDEX                   <068>
670 REM  10000    =   ENDINDEX                       <032>
680 REM  "HALLO"  =   ELEMENT                        <232>
690 REM ** BEI RUECKKEHR AUS "SE"   **               <016>
700 REM ** ENTHAELT "IN" DEN INDEX **                <046>
710 REM ** ODER (WENN DAS ELEMENT   **               <013>
720 REM ** NICHT GEFUNDEN WURDE)    **               <179>
730 REM ** DEN WERT -1              **               <114>
740 IF IN=-1 OR IN=10000 THEN 760                    <120>
750 PRINT IN,:GOTO 630                               <185>
760 PRINT:PRINT"{DOWN}ZEIT:"(TI-IT)/60"SEC
    "                                                <041>
770 PRINT"{DOWN}BITTE TASTE DRUECKEN"                <236>
780 GET T$:IF T$=""THEN 780                          <119>
790 REM                                              <090>
800 REM ****************************                 <249>
810 REM * 1.BEFEHL 'SO' MIT OPTION *                 <050>
820 REM ****************************                 <049>
830 REM                                              <130>
835 PRINT"{CLR}ERSTER BEFEHL MIT OPTION"             <147>
840 DATA NULL,ZWEI,VIER,SECHS,ACHT,ZEHN,EI
    NS,DREI,FUENF,SIEBEN,NEUN                        <044>
850 DATA 0,2,4,6,8,10,1,3,5,7,9                      <209>
860 CLR:DIM NR(10),NR$(10)                           <148>
870 REM ** EINLESEN IN FELD NR$ **                   <118>
880 FOR S=0 TO 10                                    <056>
890 :READ X$:NR$(S)=X$                               <070>
900 NEXT S                                           <046>
910 REM ** EINLESEN IN FELD NR  **                   <074>
920 FOR S=0 TO 10                                    <096>
930 :READ X:NR(S)=X                                  <224>
940 NEXT S                                           <086>
950 REM ** AUSGABE FELD VOR SORTIERUNG **            <017>
960 PRINT"{DOWN}INDEX NR$ VORHER NR * NR$
    NACHHER NR"                                      <023>
970 FOR S=0 TO 10                                    <146>
980 :PRINT S;TAB(6)NR$(S)TAB(16)NR(S)                <168>
990 NEXT S                                           <136>
1000 REM ** AUFRUF DES BEFEHL "SO" MIT OPT
     ION **                                          <037>
1010 SO,NR$,0,10,0,NR                                <081>
1020 REM SO   =   BEFEHL                             <250>
1030 REM NR$  =   FELDNAME 1                         <223>
1040 REM 0    =   ANFANGSINDEX                       <132>
1050 REM 10   =   ENDINDEX                           <155>
1060 REM 0    =   SORTIERUNGSRICHUTNG(ABST
     EIGEND)                                         <251>
1070 REM NR   =   FELDNAME 2                         <004>
1080 REM                                             <126>
1090 PRINT"{DOWN}SORTIEREN VON NR$ ABSTEIG
     END"                                            <238>
1095 PRINT"{DOWN}NR WIRD ENTSPRECHEND MITS
     ORTIERT"                                        <077>
1097 PRINT"{DOWN}BITTE TASTE DRUECKEN"               <053>
1098 GET T$:IF T$=""THEN 1098                        <177>
1100 REM ** AUSGABE FELD NACH SORTIERUNG *
     *                                               <020>
1110 PRINT"{HOME,3DOWN}";                            <140>
1120 FOR S=0 TO 10                                   <042>
1130 :PRINT TAB(22)NR$(S)TAB(33)NR(S)                <234>
1140 NEXT S                                          <032>
1150 PRINT"{6DOWN}"                                  <183>
```


Listing 1. »Exsort.demo«. (Schluß) Bitte beachten Sie die Hinweise zum Abtippen auf Seite 16.

```
programm : exsort.          c0f8 c6ac
-------------------------------------
c0f8 : a9 03 8d 08 03 a9 c1 8d   27
c100 : 09 03 60 a0 01 b1 7a c9   d2
c108 : 53 f0 03 4c e4 a7 c8 b1   30
c110 : 7a c9 45 d0 03 4c da c5   64
c118 : c9 4f d0 ef 20 73 00 20   99
c120 : 73 00 20 73 00 20 fd ae   60
c128 : 20 ec c4 20 24 c3 4c 58   36
c130 : c1 20 fd ae 20 8a ad 20   a4
c138 : f7 b7 a5 14 8d 00 c0 a5   1e
c140 : 15 8d 01 c0 20 fd ae 20   61
c148 : 8a ad 20 f7 b7 a5 14 8d   c4
c150 : 02 c0 a5 15 8d 03 c0 60   73
c158 : 20 31 c1 20 fd ae 20 9e   98
c160 : b7 8e 04 c0 e0 02 90 07   e6
c168 : a2 78 a0 c5 4c fa c4 20   17
c170 : c7 c3 8e 06 c0 8c 07 c0   8b
c178 : 8d 08 c0 a0 00 b1 7a f0   a7
c180 : 04 c9 3a d0 08 a9 00 8d   fa
c188 : 0b c0 4c a2 c1 20 fd ae   cd
c190 : 20 ec c4 20 24 c3 20 c7   cc
c198 : c3 8e 09 c0 8c 0a c0 8d   34
c1a0 : 0b c0 a9 00 8d 05 c0 a0   bb
c1a8 : 00 20 3c c5 a0 02 20 3c   93
c1b0 : c5 a9 00 8d f6 c0 8d f7   97
c1b8 : c0 a0 12 20 50 c5 90 03   cc
c1c0 : 4c ae a7 a0 10 20 50 c5   30
c1c8 : a2 10 a0 12 20 2a c5 8e   64
c1d0 : 14 c0 8c 15 c0 a2 14 a0   bd
c1d8 : f6 20 9a c4 f0 db 30 d9   80
c1e0 : ad 10 c0 8d 0c c0 ad 11   17
c1e8 : c0 8d 0d c0 ad 12 c0 8d   54
c1f0 : 0e c0 ad 13 c0 8d 0f c0   62
c1f8 : 20 ef c3 a0 06 ae 08 c0   8c
c200 : e8 8a a2 0c 20 01 c5 20   b9
c208 : 2f c4 f0 17 48 ad 04 c0   3c
c210 : d0 06 68 10 0e 4c 1b c2   35
c218 : 68 30 08 a2 0c 20 e3 c4   ca
c220 : 4c fb c1 a0 06 ae 08 c0   66
c228 : e8 8a a2 0e 20 01 c5 20   21
c230 : 2f c4 f0 17 48 ad 04 c0   64
c238 : d0 06 68 30 0e 4c 43 c2   01
c240 : 68 10 08 a2 0e 20 d1 c4   b9
c248 : 4c 23 c2 a2 0c a0 0e 20   69
c250 : 9a c4 f0 02 10 67 a0 06   94
c258 : ae 08 c0 e8 8a a2 0c 20   86
c260 : 01 c5 86 fb 84 fc a0 06   24
c268 : ae 08 c0 e8 8a a2 0e 20   9e
c270 : 01 c5 86 fd 84 fe ac 08   b8
c278 : c0 20 69 c5 ad 0b c0 f0   73
c280 : 26 a0 09 ae 0b c0 e8 8a   7e
c288 : a2 0c 20 01 c5 86 fb 84   e2
c290 : fc a0 09 ae 0b c0 e8 8a   64
c298 : a2 0e 20 01 c5 86 fd 84   fb
c2a0 : fe ac 0b c0 20 69 c5 a2   79
c2a8 : 0c 20 e3 c4 a2 0e 20 d1   14
c2b0 : c4 a2 0e a0 0c 20 9a c4   13
c2b8 : 30 03 4c fb c1 a2 0e a0   a7
c2c0 : 10 20 2a c5 8e 14 c0 8c   c9
c2c8 : 15 c0 a2 12 a0 0c 20 2a   67
c2d0 : c5 8e 16 c0 8c 17 c0 a2   44
c2d8 : 14 a0 16 20 9a c4 10 22   1a
c2e0 : a2 0c a0 12 20 9a c4 10   fd
c2e8 : 0a a0 0c 20 3c c5 a0 12   e2
c2f0 : 20 3c c5 ad 0e c0 8d 12   97
c2f8 : c0 ad 0f c0 8d 13 c0 4c   78
c300 : c8 c1 a2 10 a0 0e 20 9a   84
c308 : c4 10 0a a0 10 20 3c c5   e9
c310 : a0 0e 20 3c c5 ad 0c c0   c2
c318 : 8d 10 c0 ad 0d c0 8d 11   c2
c320 : c0 4c c8 c1 a9 04 8d 14   8a
c328 : c0 a2 00 86 03 86 04 20   bf
c330 : 73 00 f0 3f c9 3a f0 3b   70
c338 : c9 24 f0 14 c9 25 f0 18   8c
c340 : c9 2c f0 2f 15 03 95 03   07
c348 : e0 02 f0 20 e8 4c 2f c3   9f
c350 : a9 02 8d 14 c0 4c 63 c3   64
c358 : a9 01 8d 14 c0 a5 03 09   bf
c360 : 80 85 03 a5 04 09 80 85   ae
c368 : 04 4c 2f c3 a2 9e a0 c5   04
c370 : 4c fa c4 a5 2f 85 fb a5   79
c378 : 30 85 fc 4c a4 c3 a0 00   1e
```

Listing 2. »Exsort.«

```
c380 : b1 fb c5 03 d0 0a c8 b1   e4
c388 : fb c5 04 d0 03 4c b7 c3   7a
c390 : a0 02 18 a5 fb 71 fb aa   7d
c398 : 08 c8 28 a5 fc 71 fb 85   19
c3a0 : fc 8a 85 fb a5 fb c5 31   76
c3a8 : d0 d4 a5 fc c5 32 d0 ce   ba
c3b0 : a2 8f a0 c5 4c fa c4 a0   ec
c3b8 : 04 b1 fb c9 01 d0 01 60   28
c3c0 : a2 b9 a0 c5 4c fa c4 18   ff
c3c8 : a0 06 b1 fb ed 02 c0 08   59
c3d0 : 88 b1 fb 28 ed 03 c0 b0   90
c3d8 : 07 a2 00 a9 12 4c 3b a4   1f
c3e0 : 18 a5 fb 69 07 aa a5 fc   4d
c3e8 : 69 00 a8 ad 14 c0 60 18   2a
c3f0 : ad 10 c0 6d 12 c0 aa ad   b0
c3f8 : 11 c0 6d 13 c0 4a 8d 15   e6
c400 : c0 8a 6a 8d 14 c0 ae 08   64
c408 : c0 e8 8a a0 06 a2 14 20   f9
c410 : 01 c5 86 28 84 29 ad 08   f3
c418 : c0 c9 04 d0 04 8a 4c a2   e3
c420 : bb ae 08 c0 8a a8 b1 28   51
c428 : 95 61 ca 88 10 f8 60 86   89
c430 : 28 84 29 ad 08 c0 c9 02   4c
c438 : f0 09 c9 01 f0 41 8a 20   c3
c440 : 5b bc 60 a0 01 b1 28 85   6f
c448 : 03 c8 b1 28 85 04 a0 00   1c
c450 : 4c 62 c4 b1 03 d1 62 f0   5f
c458 : 08 b0 03 a9 01 60 a9 ff   68
c460 : 60 c8 98 a0 00 d1 28 f0   70
c468 : 08 c5 61 f0 04 a8 4c 53   27
c470 : c4 b1 28 c5 61 f0 05 b0   e3
c478 : e5 4c 5b c4 a9 00 60 a5   5a
c480 : 61 8d 15 c0 a5 62 8d 14   d1
c488 : c0 a0 00 b1 28 8d 17 c0   9b
c490 : c8 b1 28 8d 16 c0 a2 14   07
c498 : a0 16 bd 00 c0 d9 00 c0   0f
c4a0 : d0 0b bd 01 c0 d9 01 c0   e6
c4a8 : d0 03 4c 7c c4 bd 01 c0   5c
c4b0 : 30 17 b9 01 c0 30 a4 38   8b
c4b8 : bd 00 c0 f9 00 c0 bd 01   e4
c4c0 : c0 f9 01 c0 90 98 4c 5b   8b
c4c8 : c4 b9 01 c0 10 90 4c b7   e7
c4d0 : c4 38 bd 00 c0 e9 01 9d   ba
c4d8 : 00 c0 bd 01 c0 e9 00 9d   5e
c4e0 : 01 c0 60 fe 00 c0 d0 03   89
c4e8 : fe 01 c0 60 a5 7a 38 e9   86
c4f0 : 01 85 7a a5 7b e9 00 85   19
c4f8 : 7b 60 86 22 84 23 4c 47   aa
c500 : a4 85 28 bd 00 c0 85 71   28
c508 : bd 01 c0 85 72 a9 00 85   a6
c510 : 29 98 48 20 57 b3 86 28   19
c518 : 84 29 68 a8 18 a5 28 79   a2
c520 : 00 c0 aa a5 29 79 01 c0   c3
c528 : a8 60 38 b9 00 c0 fd 00   44
c530 : c0 48 b9 01 c0 fd 01 c0   24
c538 : a8 68 aa 60 ae 05 c0 b9   54
c540 : 00 c0 9d 19 c0 b9 01 c0   8a
c548 : 9d 88 c0 e8 8e 05 c0 60   4b
c550 : ae 05 c0 ca 10 02 38 60   bd
c558 : bd 19 c0 99 00 c0 bd 88   13
c560 : c0 99 01 c0 8e 05 c0 18   89
c568 : 60 b1 fb aa b1 fd 91 fb   3e
c570 : 8a 91 fd 88 10 f3 60 ff   75
c578 : 57 52 4f 4e 47 20 53 4f   f7
c580 : 52 54 49 4e 47 20 44 49   32
c588 : 52 45 43 54 49 4f ce 41   a5
c590 : 52 52 41 59 20 4e 4f 54   e1
c598 : 20 46 4f 55 4e c4 57 52   67
c5a0 : 4f 4e 47 20 41 52 52 41   5f
c5a8 : 59 20 4e 41 4d c5 57 52   d2
c5b0 : 4f 4e 47 20 49 4e 44 45   9f
c5b8 : d8 4f 4e 4c 59 20 4f 4e   c5
c5c0 : 45 20 44 49 4d 45 4e 53   2e
c5c8 : 49 4f 4e 20 41 52 52 41   c3
c5d0 : d9 00 ff 00 ff 00 ff 20   e9
c5d8 : ff 00 20 73 00 20 73 00   1d
c5e0 : 20 73 00 20 fd ae 20 ec   6e
c5e8 : c4 20 24 c3 20 31 c1 20   11
c5f0 : c7 c3 8e 06 c0 8c 07 c0   0b
c5f8 : 8d 08 c0 20 fd ae 20 9e   d0
c600 : ad a2 02 a0 00 20 9a c4   88
c608 : 10 07 a2 ae a0 c5 4c fa   7a
c610 : c4 ad 08 c0 c9 02 f0 14   5e
c618 : 20 8d ad ad 08 c0 c9 04   d6
c620 : f0 17 20 aa b1 85 61 84   cf
c628 : 62 4c 39 c6 20 a3 b6 85   dc
c630 : 61 a5 22 85 62 a5 23 85   88
c638 : 63 ae 08 c0 e8 8a a0 06   7e
c640 : a2 00 20 01 c5 8e 09 c0   81
c648 : 8c 0a c0 ae 09 c0 ac 0a   3d
c650 : c0 20 2f c4 f0 2a a2 00   70
c658 : a0 02 20 9a c4 f0 1a a2   d6
c660 : 00 20 e3 c4 38 ad 09 c0   98
c668 : 6d 08 c0 8d 09 c0 ad 0a   1c
c670 : c0 69 00 8d 0a c0 4c 4b   05
c678 : c6 a9 ff a0 ff 4c 86 c6   31
c680 : ad 01 c0 ac 00 c0 20 91   1d
c688 : b3 a9 00 85 0d 85 0e a9   49
c690 : 49 85 45 a9 4e 85 46 20   8d
c698 : e7 b0 a6 47 a4 48 20 d4   21
c6a0 : bb 4c ae a7 ff 00 ff ff   22
c6a8 : c9 20 f0 0d ff 00 ff 00   5f
```

Listing 2. »Exsort.«. Bitte beachten Sie bei der Eingabe den MSE auf Seite 18.

Hypra-Load

Kaffeepause! Dieses Wort kommt wohl jedem Commodore 64-Besitzer in den Sinn, der ein längeres Programm laden möchte. Es geht aber fünfmal schneller. Alles, was Sie benötigen ist das Programm »Hypra-Load« oder das neue Betriebssystem »Hypra-Perfekt«.

Das die Floppy 1541 nicht gerade die schnellste ist, ist wohl allgemein bekannt. Zumindest ein Vorgang läßt sich mit relativ geringem Aufwand stark beschleunigen: das Laden von Programmen.

Das Funktionsprinzip von Hypra-Load sieht folgendermaßen aus. Im Computer wie auch im Floppy-Laufwerk werden neue Laderoutinen installiert, die neue Busroutinen enthalten. Mit diesen neuen Busroutinen wird die Datenübertragung um den Faktor 8 bis 10 beschleunigt, so daß die Ladegeschwindigkeit auf das softwaremäßig erreichbare Maximum gebracht wird.

Eine Software-Lösung hat allerdings zwei entscheidende Nachteile: Einerseits muß vor dem schnellen Laden die neue LOAD-Routine in den Speicher gebracht werden, andererseits funktionieren manche Originalprogramme mit Kopierschutz nicht mehr. Diese Nachteile wurden bei den beiden hier vorgestellten Versionen so gut wie möglich umgangen.

»Hypra-Load 2.1« ist eine besonders komfortable Hypra-Load-Version mit integriertem Auto-Starter und Auto-Lader. Sie wurde von den Autoren des original »Hypra-Load« (aus der 64'er, Ausgabe 10/85) völlig neu geschrieben. Dies dürfte wohl die beste Lösung für diejenigen sein, die sich kein neues Betriebssystem in ein EPROM brennen wollen oder können.

»Hypra-Perfekt« ist ein solches Betriebssystem. Neben den schnellen Lade-Routinen sind hier auch eine Funktionstastenbelegung und einige andere Extras eingebaut.

Hypra-Load

Um Hypra-Load beim späteren Betrieb möglichst komfortabel zu halten, ist die Eintipparbeit etwas komplizierter. Sie benötigen als allererstes eine leere Diskette, die Sie am besten formatieren. Nun tippen Sie das Listing 1 mit dem MSE ab und speichern es auf genau dieser leeren Diskette. Sie sollten keinesfalls probieren, auf eine Diskette zu speichern, auf der schon Programme stehen, weil diese beim nächsten Schritt unrettbar verloren gehen!

Nun tippen Sie das kurze Listing 2 ab. Legen Sie die Diskette, auf der sich das gespeicherte MSE-Listing befindet, ein und starten Sie das Basic-Programm. Nach wenigen Sekunden meldet sich der Computer mit READY. Nun ist Ihre Hypra-Load-Version fertig. Um sie auf andere Disketten zu überspielen, können Sie jedes beliebige Kopierprogramm für Einzelfiles verwenden. Die Disketten, auf die Hypra-Load dann kopiert werden soll, dürfen ruhig schon Programme enthalten, es müssen aber insgesamt fünf Blöcke auf der Diskette frei sein, damit noch Platz für Hypra-Load ist.

Wie arbeitet man nun mit Hypra-Load? Es gibt zwei Betriebsarten: In der normalen Betriebsart lädt man Hypra-Load mit LOAD":*",8,1. Daraus folgt, daß es das erste Programm auf der Diskette sein muß! Hypra-Load installiert sich dann vollautomatisch, ohne daß RUN eingetippt werden muß, und meldet sich mit den Namen der Autoren. Ab sofort wird von Diskette fünfmal schneller geladen. Zusätzlich sind als Standardwerte für den LOAD-Befehl »,8,1« schon fest vorgegeben. »LOAD"name"« lädt also immer von Diskette.

Bei einem Reset schaltet sich Hypra-Load ab, ebenso wenn wichtige Speicherbereiche überschrieben werden und somit ein Systemabsturz provoziert wird.

Viel interessanter und komfortabler, Hypra-Load zu nutzen, ist allerdings die zweite Methode. Wenn Hypra-Load sich zusammen mit anderen Programmen auf einer Diskette befindet, genügt der Befehl »LOAD"↑*name",8,1« um mehrere Vorgänge gleichzeitig in Gang zu setzen: Hypra-Load wird geladen und gestartet, danach wird vollautomatisch das Programm mit dem Namen »name« schnell nachgeladen und ebenfalls gestartet. Voraussetzung dabei ist, daß die Hypra-Load-Version auf der Diskette genau mit dem im MSE-Listing angegebenen Namen (»↑Hypra-Load«) gespeichert ist.

Einige Anmerkungen zum Auto-Lader: Nach der Installation von Hypra-Load wird überprüft, ob der aktuelle Filename mit einem "↑" (Pfeil nach oben) beginnt. Dann werden die ersten beiden Buchstaben vom Filenamen abgetrennt und das gewünschte Programm nachgeladen. Nach dem Laden desaktiviert sich Hypra-Load sofort und startet das geladene Programm durch Simulation des RUN-Befehls. Mehrteilige Programme laufen also ab dem zweiten Teil in der normalen Ladegeschwindigkeit ab. Für die meisten Anwendungsfälle ist diese Lösung allerdings vollkommen befriedigend.

Diese Hypra-Load-Version läuft unserer Erfahrung nach sowohl auf C 64- wie auch SX 64-Computern und dem C 128 im 64-er Modus. Als Laufwerke sind die 1541 und die 1571 von Commodore verwendbar.

Der Bildschirm bleibt während des Ladevorgangs ausgeschaltet, da sonst der VIC-Chip die schnellen Bus-Routinen aus dem Takt bringen würde. Das Blinken der roten Laufwerks-LED ist völlig normal, es zeigt jeweils einen übertragenen Programmblock an. Damit läßt sich optisch die Arbeit von Hypra-Load verfolgen.

Weitere, am seriellen Bus angeschlossene, Geräte dürften die Übertragung nicht stören. Solange Hypra-Load aktiviert ist, führen Kassetten-Zugriffe zum System-Absturz!

Soll Hypra-Load abgeschaltet werden, genügt die Befehlsfolge »POKE 1,55«.

Der VERIFY-Befehl funktioniert nicht bei aktiviertem Hypra-Load.

Für diejenigen, die sich für die Programmierung des Autostarts von Hypra-Load interessieren: Er wurde durch Verbiegen des Stop-Vektors realisiert. (Karsten Schramm/bs)

Hypra-Perfekt

Unsere zweite Hypra-Load-Version ist eine Betriebssystem-Version. Das Original-Betriebssystem des Commodore 64 wird durch ein modifiziertes Betriebssystem ersetzt, welches die schnellen Lade-Routinen sowie eine Funktionstastenbelegung enthält.

Somit muß das Kernal-ROM ausgetauscht werden. Als neuer Programmträger dient ein 2764-EPROM, dessen Inhalt mit dem nachfolgend abgedruckten Programm (Listing 3) erzeugt wird. Wie das von statten geht, läßt sich mit wenigen Worten erklären. Das Betriebssystem (Kernal) des Commodore 64 befindet sich im Speicherbereich von $E000 bis $FFFF und hat somit die Länge von 8 KByte. Das gesamte Kernal als MSE-Listing abzudrucken würden den Rahmen des Sonderheftes sprengen. Dies wäre auch gar nicht notwendig, da ja nur Teile des Betriebssystems geändert werden. Deswegen haben wir die Form des, dank des MSEs etwas altmodischen, aber hier angebrachten DATA-Laders gewählt.

Das abgedruckte Programm kopiert zunächst das Kernal in den Speicherbereich von $6000 bis $7FFF. Wer einen anderen Speicherbereich verwenden möchte, muß die Leseschleife in Zeile 30 on den Offset-Wert in Zeile 15 (OF) verändern.

Nach dem Kopieren befindet sich ein originalgetreues Abbild des Kernals im RAM. Der nächste Schritt, der automatisch vom Programm durchgeführt wird, ist die Veränderung der einzelnen Speicherstellen. Ist dieser Vorgang beendet, steht unser neues Kernal von $6000 bis $7FFF im Speicher und sollte sofort mit einem Monitor abgespeichert werden. Danach kann man es dann auf ein 2764-EPROM brennen.

Bis hierhin war alles relativ einfach, jetzt aber kommt die schwierigste Hürde auf dem Weg zum neuen Betriebssystem. Ein Adaptersockel muß gebaut werden, denn leider sind die Pin-Belegungen des Original-Kernals und des 2764-EPROMS nicht identisch. Wer sich diese Arbeit nicht machen möchte, der kann diesen Adaptersockel im 64'er-Hardwareservice demnächst auch kaufen. Sollten Sie schon eine Kernal-Umschaltplatine besitzen, wie wir sie ebenfalls anbieten werden, entfällt der Bastelaufwand für Sie ebenfalls.

Wer sich nun den Sockel selber bauen will, oder muß, der besorge sich einen 24poligen und einen 28poligen Sockel. Das Bild 1 zeigt, welche Pins der Sockel miteinander verbunden werden müssen. Nach der Fertigstellung sieht das Ganze wie ein Doppeldecker-Bus mit überhängendem Dach aus. Im oberen Stockwerk dieses Busses nimmt nun der 2764 mit dem neuen Betriebssystem seinen Platz ein. Die 24 Pins des unteren Fahrwerks finden im Steckplatz U4 des Computers ihre Heimat. Das überhängende Dach zeigt, zusammen mit der Kerbe des EPROMs, in Richtung des User-Ports.

Sollte sich der Computer nach dem Einschalten mit geändertem Farbbild, aber sonst gewohnter Anzeige melden, haben Sie es geschafft. Ihnen steht nun eine recht leistungsfähige Hypra-Load-Version zur Verfügung. Neben dem Schnelllade-System wurden noch die Funktionstasten belegt. Die Belegung können Sie Bild 2 entnehmen. Das Laden eines Directorys zerstört nicht mehr das aktuelle Programm im Speicher.

Um die Kompatibilität zu fertigen Softwareprodukten möglichst groß zu halten, wurde auch hier eine Abschaltautomatik installiert. Sollte schnelles Laden nicht mehr möglich sein, wird auf langsames Laden umgestellt. Wenn diese Automatik versagt, kann das schnelle Laden auch von Hand abgeschaltet werden. Die Speicherstelle 2 ist zum Steuerregister umfunktioniert worden. Die einzelnen Funktionen von Hypra-Perfekt lassen sich folgendermaßen schalten:

POKE 2,0: Grundzustand von Hypra-Perfekt, alles eingeschaltet

28poliger Sockel	24poliger Sockel
1,28,27,26	24
2	21
3	1
4	2
5	3
6	4
7	5
8	6
9	7
10	8
11	9
12	10
13	11
14,20	12
15	13
16	14
17	15
18	16
19	17
20,14	18
21	19
22	20
23	18
24	22
25	23
1,28,27,26	24

Pin 1 eines jeden Sockels ist links von der Kerbe, Pin 24 beziehungsweise 28 ist rechts davon.

Bild 1. Das Verbindungsschema für den Adaptersockel

POKE 2,32: Funktionstasten abschalten
POKE 2,64: Directory wieder normal laden
POKE 2,128: Schnelles Laden abschalten.

Natürlich sind auch Kombinationen möglich, so zum Beispiel »POKE 2,32+64+128« schaltet alle Funktionen ab.

Besitzer eines Hypra-Perfekt-Kernals müssen, sofern Sie keine Umschaltplatine besitzen, leider auf die Kassetten-Routinen verzichten. Die RS232-Schnittstelle ist aber voll erhalten geblieben.

Wer seine individuelle Farbkombination beim Einschalten haben möchte, braucht nur in den Zeilen 50 bis 70 die Farbwerte zu ändern.

Und viel Spaß mit »Hypra-Perfekt«.

(Ernst Schöberl/aw/bs)

F1 = LOAD"$",8 ohne Programmverlust
F2 = LOAD
F3 = LIST
F4 = SAVE
F5 = RUN
F6 = SYS64738
F7 = OPEN1,8,15,
F8 = CLOSE1
SHIFT & RUN/STOP lädt und startet das erste Programm von der Diskette

Bild 2. Die Belegung der Funktionstasten

```
PROGRAMM : ↑HYPRA-LOAD    1000 14BC
-----------------------------------
1000 : A9 08 85 AF A9 00 85 AE  13
1008 : A9 6A 8D 2C 03 A9 0B 8D  94
1010 : 2D 03 A9 ED 8D 28 03 A9  60
1018 : F6 8D 29 03 60 86 AE 00  75
1020 : 00 00 00 4C 48 B2 00 31  26
1028 : EA 66 FE 47 FE 4A F3 91  23
1030 : F2 0E F2 50 F2 33 F3 57  37
1038 : F1 CA F1 ED 02 A9 BB A2  6A
1040 : EE 85 A7 86 A8 A9 00 A2  C9
1048 : 03 85 A9 86 AA A9 08 20  A1
1050 : 0C ED A9 6F 20 B9 ED A9  86
1058 : 4D 20 DD ED A9 2D 20 DD  2B
1060 : ED A9 57 20 DD ED A0 00  CB
1068 : A5 A9 20 DD ED A5 AA 20  9D
1070 : DD ED A9 1D 20 DD ED B1  5E
1078 : A7 20 DD ED C8 C0 1D 90  8C
1080 : F6 20 FE ED 18 A5 A7 69  24
1088 : 1D 85 A7 90 03 E6 A8 18  9E
1090 : A5 A9 69 1D 85 A9 90 02  F4
1098 : E6 AA A5 A9 C9 D0 D0 AD  34
10A0 : A9 08 20 0C ED A9 6F 20  01
10A8 : B9 ED A9 4D 20 DD ED A9  68
10B0 : 2D 20 DD ED A9 45 20 DD  23
10B8 : ED A9 01 20 DD ED A9 04  BA
10C0 : 20 DD ED 20 FB ED 20 B3  65
10C8 : EE 20 B3 EE 78 A9 0B 8D  AD
10D0 : 11 D0 A9 00 85 BB A9 FE  8F
10D8 : 85 BC 20 39 F8 C9 FF F0  AA
10E0 : 61 20 39 F8 85 B7 20 39  C8
10E8 : F8 A8 A5 B7 D0 03 88 84  E5
10F0 : BC A5 BB D0 0A C6 BC C6  DF
10F8 : BC 20 39 F8 20 39 F8 A0  23
1100 : 00 20 39 F8 91 AE C8 C4  B9
1108 : BC D0 F6 18 A5 BC 65 AE  20
1110 : 85 AE 90 02 E6 AF E6 BB  50
1118 : A5 B7 D0 BA C6 BB D0 04  BA
1120 : E6 BC E6 BC A5 BC C9 FE  1B
1128 : F0 08 20 39 F8 E6 BC 18  35
1130 : 90 F2 A9 00 2C A9 1D 85  33
1138 : 90 4C 85 F8 A5 90 C9 01  77
1140 : 48 08 98 A6 AE A4 AF 4C  EF
1148 : 61 00 A9 23 8D 00 DD 2C  21
1150 : 00 DD 50 FB A9 03 8D 00  BB
1158 : DD A2 07 CA D0 FD AD 00  55
1160 : DD 2A 2A 66 B0 6A 66 B0  03
1168 : EA EA AD 00 DD 2A 2A 66  D7
1170 : B0 6A 66 B0 EA EA AD 00  C2
1178 : DD 2A 2A 66 B0 6A 66 B0  1B
1180 : EA EA AD 00 DD 2A 2A 66  EF
1188 : B0 6A 66 B0 A5 B0 49 FF  22
1190 : A2 03 8E 00 DD 60 68 A8  2B
1198 : 68 AA 48 98 48 E0 E0 90  AB
11A0 : 0D A9 00 A8 59 00 A0 C8  41
11A8 : D0 FA C9 80 D0 10 A9 00  AC
11B0 : A8 59 BB EE C8 D0 FA C9  64
11B8 : 70 D0 03 A0 35 2C A0 37  0B
11C0 : A9 85 8D 61 00 A9 01 8D  28
11C8 : 62 00 A9 28 8D 63 00 A9  E1
11D0 : 68 8D 64 00 A9 60 8D 65  B7
11D8 : 00 A9 1B 8D 11 D0 4C 2B  44
11E0 : F8 A5 00 29 06 C9 02 F0  69
11E8 : 03 4C 9E FD EA A9 05 85  94
11F0 : 09 A2 5A 86 4B A2 00 A9  CF
11F8 : 52 85 24 20 56 F5 50 FE  6E
1208 : C6 4B D0 EF A9 0A 4C 69  95
1210 : F9 50 FE B8 AD 01 1C 95  86
1218 : 25 E8 E0 07 D0 F3 20 97  27
1220 : F4 A5 16 45 17 45 18 45  9C
1228 : 19 45 1A F0 07 C6 09 D0  F5
1230 : C0 4C 1E F4 A5 18 C5 06  7B
1238 : F0 03 4C 0B F4 85 22 A9  76
1240 : 06 85 31 4C EB 05 [illegible] 12  70
1248 : A6 13 85 16 86 17 A5 06  60
1250 : 85 18 A5 07 85 19 A9 00  F3
1258 : 45 16 45 17 45 18 45 19  39
1260 : 85 1A 20 34 F9 A2 5A 20  DF
1268 : 56 F5 A0 00 50 FE B8 AD  1C
1270 : 01 1C D9 24 00 F0 06 CA  AF
1278 : D0 ED 4C 51 F5 C8 C0 08  35
1280 : D0 EA 20 56 F5 50 FE B8  E7
1288 : AD 01 1C 91 30 C8 D0 F5  68
1290 : A0 BA 50 FE B8 AD 01 1C  B6
1298 : 99 00 01 C8 D0 F4 20 E0  82
12A0 : F8 A5 38 C5 47 F0 03 4C  D2
12A8 : F6 F4 20 E9 F5 C5 3A F0  B6
12B0 : 03 4C 02 F5 A0 00 A9 55  74
12B8 : 20 74 04 B9 00 06 20 74  E4
12C0 : 04 C8 D0 F7 AD 00 1C 49  39
12C8 : 08 8D 00 1C AD 00 06 D0  AF
12D0 : 03 4C 9E FD C5 18 D0 F9  B5
12D8 : 85 06 AD 01 06 85 07 4C  2D
12E0 : 65 03 78 A9 08 8D 00 18  37
12E8 : A5 18 8D 00 06 85 06 A5  ED
12F0 : 19 8D 01 06 85 07 A9 04  10
12F8 : 85 78 A9 E2 20 6B 04 C9  81
1300 : 02 90 33 A0 00 84 78 A4  7B
1308 : 78 B9 DB FE F0 12 58 20  75
1310 : 76 D6 78 A9 E2 20 6B 04  29
1318 : C9 02 90 1A E6 78 D0 E7  8F
1320 : A9 C0 20 6B 04 A9 E2 20  F8
1328 : 6B 04 C9 02 90 08 A9 FF  38
1330 : 20 74 04 4C 22 EB AD 00  4D
1338 : 06 F0 F8 C5 18 F0 C4 AD  25
1340 : 00 06 85 06 AD 01 06 85  6B
1348 : 07 4C 15 04 85 00 58 A5  40
1350 : 00 30 FC 78 60 85 77 A2  0C
1358 : 01 8A 2C 00 18 F0 FB A9  F6
1360 : 00 8D 00 18 8A 2C 00 18  64
1368 : D0 FB A2 00 8A 66 77 2A  ED
1370 : 2A 66 77 2A 2A 8D 00 18  30
1378 : 8A 66 77 2A 2A 66 77 2A  61
1380 : 2A 8D 00 18 8A 66 77 2A  82
1388 : 2A 66 77 2A 2A 8D 00 18  48
1390 : 8A 66 77 2A 2A 66 77 2A  79
1398 : 2A 8D 00 18 A2 02 CA D0  93
13A0 : FD A9 08 8D 00 18 60 A0  A9
13A8 : 00 84 61 A9 A0 85 62 B1  9B
13B0 : 61 91 61 E6 61 D0 F8 E6  5D
13B8 : 62 F0 0C A5 62 C9 C0 D0  63
13C0 : EE A9 E0 85 62 D0 E8 A9  0F
13C8 : E5 8D D6 FD A9 4C 8D F9  10
13D0 : F4 A9 2C 8D FA F4 A9 F7  43
13D8 : 8D FB F4 A2 00 BD 00 08  F2
13E0 : 9D 2C F7 E8 D0 F7 BD 00  72
13E8 : 09 9D 2C F8 E8 E0 A4 D0  B4
13F0 : F5 A2 00 BD A4 09 9D BB  6F
13F8 : EE E8 D0 F7 BD A4 0A 9D  F2
1400 : BB EF E8 E0 C6 D0 F5 A9  27
1408 : 08 8D DA E1 A9 01 8D DC  5C
1410 : E1 A9 35 85 01 A0 00 B1  3C
1418 : BB C9 5E F0 59 20 53 E4  1B
1420 : 20 BF E3 18 20 15 FD A9  12
1428 : F7 A0 0B 20 1E AB 20 44  7F
1430 : A6 4C 9D E3 0D 48 59 50  F9
1438 : 52 41 2D 4C 4F 41 44 20  50
1440 : 56 32 2E 31 20 28 43 29  04
1448 : 20 31 39 38 35 20 54 52  A1
1450 : 49 42 41 52 0D 42 4F 52  1A
1458 : 49 53 20 53 43 48 4E 45  F8
1460 : 49 44 45 52 2B 4B 41 52  1E
1468 : 53 54 45 4E 20 53 43 48  3B
1470 : 52 41 4D 4D 0D 00 18 A9  E4
1478 : 02 65 BB 85 BB 90 02 E6  E2
1480 : BC C6 B7 C6 B7 EA A2 00  C3
1488 : BD 59 0C 9D C0 02 E8 E0  2A
1490 : 25 D0 F5 4C C0 02 20 53  67
1498 : E4 20 BF E3 18 20 15 FD  CB
14A0 : 20 44 A6 20 D5 FF A9 37  02
14A8 : 85 01 A5 AE 85 2D A5 AF  A5
14B0 : 85 2E 20 63 A6 20 8E A6  B4
14B8 : 4C AE A7 00 00 00 00 00  45
```

Listing 1. »Hypra-Load 2.1«. Bitte beachten Sie die Hinweise im Text und die Eingabehinweise auf Seite 18.

```
10 OPEN 1,8,15
20 OPEN 2,8,2,"#"
30 PRINT#1,"U1 2 0 17 0"
40 PRINT#1,"B-P 2 2"
50 PRINT#2,CHR$(237);CHR$(2);
60 PRINT#1,"U2 2 0 17 0"
70 CLOSE 2,1

READY.
```

Listing 2. Hilfsprogramm zur Erstellung von »Hypra-Load 2.1«. Bitte beachten Sie die Hinweise im Text!

```
10 POKE 56,96:POKE 55,0:CLR                   <236>
15 OF=32768                                   <220>
20 PRINT"{CLR,3DOWN}KOPIEREN DER ROM-DATEN
   NACH $6000":PRINT                          <209>
30 FOR I=6*4096 TO 8*4096-1:POKE I,PEEK(I+
   OF):NEXT I                                 <028>
40 REM EINSCHALTFARBEN SETZEN                 <160>
50 POKE 58677-OF,5: REM ZEICHEN GRUEN         <172>
60 POKE 60633-OF,0: REM RAHMEN SCHWARZ        <248>
70 POKE 60634-OF,0: REM HINTERGR. SCHWARZ     <216>
80 PRINT"LESEN DER DATA-ZEILEN":PRINT:PRIN
```

Listing 3. Programm zur Erstellung von »Hypra-Perfekt«

```
   T                                            <184>
100 T=0                                         <077>
110 T=T+1:READ A:IF A=0 THEN 240                <071>
120 READ B: REM ANZAHL DER BYTES                <099>
130 READ P1: REM PRUEFSUMME                     <231>
140 P2=0:PRINT"BLOCK ";T;"{2SPACE}";            <018>
150 FOR I=A-OF TO A-OF-1+B                      <039>
160 READ D:POKE I,D:P2=P2+D                     <193>
170 NEXT I                                      <254>
180 IF P2<>P1 THEN 210                          <101>
190 PRINT"{3SPACE}OK"                           <026>
200 GOTO 110                                    <144>
210 PRINT"PRUEFSUMME FALSCH: ";P2;" STATT
    ";P1:PRINT                                  <179>
220 GET A$:IF A$=""THEN 220                     <253>
230 GOTO 110                                    <174>
240 PRINT:PRINT"FERTIG!":END                    <180>
8000 REM *** BLOCK 1: $EB3F-$EB41 ***           <201>
8001 DATA 60223,3,359                           <001>
8002 DATA 76,32,251                             <001>
8005 REM *** BLOCK 2: $FB20-$FB68 ***           <253>
8006 DATA 64288,73,10154                        <240>
8007 DATA 232,134,198,201,133,144,4,201,14
     1,144,3,76,66,235,157,119,2,72,152         <171>
8008 DATA 72,160,0,169,32,37,2,208,13,185,
     162,251,221,119,2,240,11,200,192           <145>
8009 DATA 224,208,243,104,168,104,76,66,23
     5,200,185,162,251,201,133,144,4,201        <254>
8010 DATA 141,144,238,236,137,2,176,233,15
     7,119,2,232,134,198,76,79,251              <015>
8015 REM *** BLOCK 3: $FBA0-$FBDF ***           <169>
8016 DATA 64416,64,5759                         <104>
8017 DATA 252,255,133,76,207,34,36,34,44,5
     6,13,137,76,79,65,68,134,76,73,83          <087>
8018 DATA 84,13,138,83,65,86,69,135,82,85,
     78,13,139,83,121,54,52,55,51,56,13         <062>
8019 DATA 136,79,208,49,44,56,44,49,53,44,
     34,140,67,76,79,83,69,49,13,136,255        <004>
8020 DATA 255,255                               <237>
8100 REM *** BLOCK 4: $E5EF-$E5F5 ***           <180>
8101 DATA 58862,8,1523                          <004>
8102 DATA 32,235,240,134,198,189,255,240        <244>
8105 REM *** BLOCK 5: $F0D9-$F0EA ***           <154>
8106 DATA 61657,45,5377                         <082>
8107 DATA 76,207,34,58,42,34,44,56,44,49,5
     8,88,32,8,249,76,40,245,162,0,189          <055>
8108 DATA 216,240,32,210,255,232,224,12,20
     8,245,162,6,120,96,234,255,255,255         <123>
8109 DATA 13,82,85,78,58,13                     <062>
8110 REM *** BLOCK 6: $F4AD-$F4B7 ***           <118>
8111 DATA 62637,11,1586                         <145>
8112 DATA 201,4,176,7,169,8,133,186,234,23
     4,234                                      <159>
8115 REM *** BLOCK 7: $F4F9-$F500 ***           <139>
8116 DATA 62713,8,820                           <213>
8117 DATA 165,2,48,4,76,44,247,234              <245>
8120 REM *** BLOCK 8: $F5EF-$F5F9 ***           <058>
8121 DATA 62959,11,1586                         <126>
8122 DATA 201,4,176,7,169,8,133,186,234,23
     4,234                                      <169>
8125 REM *** BLOCK 9: $F72C-$F7FF ***           <246>
8126 DATA 63276,212,31985                       <179>
8127 DATA 160,0,177,187,201,36,208,63,76,2
     24,251,255,255,255,255,255,255,255         <208>
8128 DATA 255,165,186,32,12,237,169,111,32
     ,185,237,169,77,32,221,237,169,45          <032>
8129 DATA 32,221,237,169,69,32,221,237,169
     ,69,32,221,237,169,193,32,221,237          <060>
8130 DATA 32,254,237,32,239,237,76,21,249,
     255,255,255,255,255,255,255,255,169        <112>
8131 DATA 28,162,249,133,167,134,168,169,0
     ,162,3,133,169,134,170,165,186,32          <069>
8132 DATA 12,237,169,111,32,185,237,169,77
     ,32,221,237,169,45,32,221,237,169          <060>
8133 DATA 87,32,221,237,160,0,165,169,32,2
     21,237,165,170,32,221,237,169,30           <068>
8134 DATA 32,221,237,177,167,32,221,237,20
     0,192,30,144,246,32,254,237,24,165         <096>
8135 DATA 167,105,30,133,167,144,3,230,168
     ,24,165,169,166,170,105,30,133,169         <164>
8136 DATA 144,2,230,170,224,5,144,173,201,
     0,144,169,165,186,32,12,237,169,111        <175>
8137 DATA 32,185,237,169,77,32,221,237,169
     ,45,32,221,237,169,69,32,221,237           <112>
8138 DATA 169,139,32,221,237,169,4,32,221,
     237,173,0,221                              <058>
8140 REM *** BLOCK 10: $F800-$F8FF ***          <198>
8141 DATA 63488,256,38681                       <173>
8142 DATA 133,171,169,239,45,17,208,141,17
     ,208,32,128,248,32,152,248,32,63           <167>
8143 DATA 247,8,72,166,174,164,175,104,40,
     88,96,32,39,248,201,255,240,248,160        <139>
8144 DATA 0,96,169,39,141,0,221,44,0,221,8
     0,251,169,3,141,0,221,162,7,202,208        <083>
8145 DATA 253,234,173,0,221,42,42,102,176,
     106,102,176,234,234,173,0,221,42           <227>
8146 DATA 42,102,176,106,102,176,234,234,1
     73,0,221,42,42,102,176,106,102,176         <143>
8147 DATA 234,234,173,0,221,42,42,102,176,
     106,102,176,165,176,73,255,96,234          <200>
8148 DATA 96,234,234,234,234,234,234,234,2
     34,234,234,234,234,234,234,234,169         <128>
8149 DATA 3,141,0,221,32,251,237,138,162,2
     55,234,234,234,202,208,252,170,96          <180>
8150 DATA 234,234,234,234,234,120,169,1,13
     3,167,160,255,32,29,248,192,255,208        <063>
8151 DATA 3,76,228,240,32,39,248,133,169,3
     2,39,248,133,168,165,167,240,30,32         <193>
8152 DATA 39,248,32,39,248,169,252,234,234
     ,164,169,208,30,198,168,198,168,198        <015>
8153 DATA 168,169,254,24,229,168,133,170,7
     6,229,248,165,169,208,8,198,168,169        <224>
8154 DATA 0,76,205,248,44,169,254,133,168,
     160,0,32,39,248,145,174,230,174,208        <206>
8155 DATA 2,230,175,198,168,208,241,162,0,
     134,167,165,169,208,159,198,170            <203>
8160 REM *** BLOCK 11: $F900-$F9FF ***          <043>
8161 DATA 63744,256,30043                       <025>
8162 DATA 240,6,32,39,248,76,254,248,169,1
     6,13,17,208,141,17,208,165,171,141         <115>
8163 DATA 0,221,169,64,133,144,24,96,255,1
     65,0,41,6,201,2,240,3,76,158,253           <204>
8164 DATA 234,169,5,133,9,162,90,134,75,16
     2,0,169,82,133,36,32,86,245,80,254         <177>
8165 DATA 184,173,1,28,197,36,240,9,198,75
     ,208,239,169,10,76,105,249,80,254          <121>
8166 DATA 184,173,1,28,149,37,232,224,7,20
     8,243,32,151,244,165,22,69,23,69           <247>
8167 DATA 24,69,25,69,26,240,7,198,9,208,1
     92,76,30,244,165,24,197,6,240,3,76         <007>
8168 DATA 11,244,133,34,169,6,133,49,76,92
     ,4,165,18,166,19,133,22,134,23,165         <139>
8169 DATA 6,133,24,165,7,133,25,169,0,69,2
     2,69,23,69,24,69,25,133,26,32,52           <242>
8170 DATA 249,162,90,32,86,245,160,0,80,25
     4,184,173,1,28,217,36,0,240,6,202          <162>
8171 DATA 208,237,76,81,245,200,192,8,208,
     234,32,86,245,80,254,184,173,1,28          <035>
8172 DATA 145,48,200,208,245,160,186,80,25
     4,184,173,1,28,153,0,1,200,208,244         <092>
8173 DATA 32,224,248,165,56,197,71,240,3,7
     6,246,244,32,233,245,197,58,240,3          <015>
8174 DATA 76,2,245,160,0,169,85,32,232,3,1
     [illegible]5,0,6,32,232,3,200,208,247,240  <142>
8180 REM *** BLOCK 12: $FA00-$FAFF ***          <089>
8181 DATA 64000,256,29911                       <249>
8182 DATA 111,234,234,234,133,119,162,1,13
     8,44,0,24,240,251,169,0,141,0,24           <108>
8183 DATA 138,44,0,24,208,251,162,0,138,10
     2,119,42,42,102,119,42,42,141,0,24         <137>
8184 DATA 138,102,119,42,42,102,119,42,42,
     141,0,24,138,102,119,42,42,102,119         <068>
8185 DATA 42,42,141,0,24,138,102,119,42,42
     ,102,119,42,42,141,0,24,162,2,202          <025>
8186 DATA 208,253,169,8,141,0,24,96,234,16
     9,8,141,0,24,96,234,234,234,234,234        <010>
8187 DATA 234,234,234,234,234,234,234,234,
     234,234,234,234,234,234,173,0,28           <075>
8188 DATA 9,8,141,0,28,173,0,6,208,3,76,15
     8,253,197,24,208,249,133,6,173,1           <003>
8189 DATA 6,133,7,76,101,3,234,234,234,234
     ,234,234,234,234,234,234,234,234           <169>
8190 DATA 234,234,96,96,133,0,88,165,0,48,
     252,120,96,120,169,8,141,0,24,234          <153>
8191 DATA 165,24,141,0,6,133,6,165,25,141,
     1,6,133,7,169,4,133,120,169,226,32         <201>
8192 DATA 130,4,201,2,144,51,160,0,132,120
     ,164,120,185,219,254,240,18,88,32          <212>
8193 DATA 118,214,120,169,226,32,130,4,201
     ,2,144,26,230,120,208,231,169,192          <013>
8194 DATA 32,130,4,169,226,32,130,4,201,2,
     144,8,169,255,32,232,3,76,34,235           <144>
8195 DATA 173,0,6,240                           <091>
8200 REM *** BLOCK 13: $FB00-$FB1F ***          <121>
8201 DATA 64256,32,4506                         <195>
8202 DATA 248,197,24,240,196,173,0,6,133,6
     ,173,1,6,133,7,76,160,4,234,234,234        <111>
8203 DATA 234,160,0,185,25,244,153,255,255
     ,255,255                                   <166>
8210 REM *** BLOCK 14: $FBE0-$FC3F ***          <064>
8211 DATA 64480,96,12568                        <079>
8212 DATA 165,2,41,64,240,3,76,1,245,169,1
     3,32,210,255,234,169,0,133,144,160         <203>
8213 DATA 2,132,169,32,19,238,133,170,32,2
     25,255,208,3,76,51,246,164,144,208         <195>
8214 DATA 47,32,19,238,164,144,208,40,164,
     169,136,208,225,166,170,32,205,189         <242>
8215 DATA 169,32,32,210,255,32,19,238,166,
     144,208,18,170,240,6,32,210,255,76         <245>
8216 DATA 30,252,169,13,32,210,255,160,2,2
     08,190,32,66,246,166,45,164,46,96          <067>
8217 DATA 255                                   <228>
9000 DATA 0                                     <253>
9010 REM **** ENDE ****                         <199>
```

© 64'er

Listing 3. (Schluß)

Strubs — ein Precompiler für Basic-Programme

Bei Strubs, das steht für »strukturiertes Basic«, handelt es sich um einen sogenannten Precompiler, ein Programm, welches Programmtexte mit gewissen zusätzlichen Befehlen in normale, auf jedem Commodore 64 oder VC 20 ablauffähige Programme übersetzt.

Das Programm Strubs wurde ursprünglich zu einer Zeit entwickelt, als Begriffe wie Forth oder Pascal noch Fremdworte für den C 64 waren. Der Zweck war, die Entwicklung von Programmen übersichtlicher, effizienter und bequemer zu gestalten.

Strubs bietet neue Basicbefehle

Auf der einen Seite ermöglicht es Strubs, auf sanftem Weg, das heißt im Rahmen des gewohnten Basic (aber ohne auf unübersichtliche Klimmzüge innerhalb des Commodore Basic angewiesen zu sein), also ohne gleich eine neue Programmiersprache lernen zu müssen, mit der Technik strukturierter Programmierung vertraut zu werden. Auf der anderen Seite ermöglicht es Strubs, sich mit der Arbeit mit Compilern vertraut zu machen.

Schließlich bietet die Form des Precompilers noch erhebliche Geschwindigkeitsvorteile gegenüber vergleichbaren Interpretererweiterungen. Um diese letzten beiden Punkte zu verstehen, ist es angebracht, auf die unterschiedlichen Arbeitsweisen von Interpretern und Compilern einzugehen.

Bekanntlich versteht der eigentliche Computer, das heißt hier der Microprozessor, nur die sogenannte Maschinensprache. Da diese aber extrem problemfern und unübersichtlich ist, hat man verschiedene höhere Programmiersprachen erfunden, um dem Programmierer seine Arbeit zu erleichtern. Damit aber ein in einer solchen Sprache geschriebenes Programm vom Computer verarbeitet werden kann, muß zunächst eine Übertragung wiederum von Programmen vorgenommen und zwar von Compilern oder von Interpretern. Diese beiden Programmarten unterscheiden sich grundlegend in ihrer Arbeitsweise.

Ein Interpreter besteht im wesentlichen aus einer Reihe von in Maschinensprache geschriebenen Unterprogrammen, einer Tabelle, welche die erlaubten Befehle und die Adresse des zu jedem Befehl gehörenden Unterprogramms enthält, und schließlich der Variablenverwaltung sowie der sogenannten Interpreterschleife.

Diese Schleife geht den Programmtext Schritt für Schritt durch. Zu jedem Befehl sucht sie in der Tabelle die zugehörige Unterprogrammadresse, ruft dieses Unterprogramm auf, holt den nächsten Befehl und so weiter, bis das Programmende erreicht ist. Man sieht also, daß ein großer Teil der Arbeit eines Interpreters im Suchen besteht: Suchen in der Befehlstabelle, Suchen in der Variablentabelle und nicht zuletzt Suchen nach Sprungzielen im zu interpretierenden Programm.

Diese ewige Sucherei führt nun dazu, daß Programme nur relativ langsam abgearbeitet werden. Eine Interpretererweiterung (wie etwa Simons Basic) stellt nun einfach zusätzliche Befehlsroutinen zur Verfügung und erweitert die Befehlstabelle um die neuen Befehle und Adressen. Durch diese Erweiterung der Befehlstabellen wird jetzt aber leider auch der Zeitaufwand für das Suchen größer, so daß die Programme noch langsamer als bisher schon ablaufen. Simons Basic demonstriert dies sehr anschaulich. Ein Beispiel für eine Interpretererweiterung werden wir weiter unten besprechen.

Nehmen wir zur Illustration der Arbeitsweise eines Interpreters eine Programmzeile wie die folgende:

FOR I = 0 TO 999: PRINT I: NEXT

Der Interpreter muß hier 1000mal die Befehlstabellen nach dem Befehl PRINT und 1000mal die Variablentabelle nach der Variablen I durchsuchen.

Compiler kontra Interpreter

Völlig anders arbeitet der Compiler: Er übersetzt ein Programm, das in einer Sprache geschrieben ist, welche nur der Programmierer versteht (dieses Programm nennt man Quellprogramm), in ein äquivalentes Programm (das Objektprogramm), das (meist nur noch) die Maschine versteht. Diese beiden Begriffe — Quellprogramm und Objektprogramm — sollten wir uns gut merken; sie werden noch öfter auftauchen.

Der größte Teil der Sucharbeit kann nun ein für allemal bei der Übersetzung vom Compiler geleistet werden. Die benötigten Adressen der Befehlsroutinen, der Variablen und der Sprungziele sind für immer fest in das Objektprogramm eingebaut. Dadurch können compilierte Programme oft bis zu zehn- oder mehrmal schneller sein als entsprechende Interpreterprogramme.

Diesem beträchtlichen Gewinn an Geschwindigkeit steht allerdings ein nicht minder bedeutender Verlust an Bequemlichkeit gegenüber. Zum einen erfordert selbst die geringste Programmänderung eine vollständige Neuübersetzung des Programms. Dies allein kann bei umfangreichen Programmen erhebliche Zeit beanspruchen, zumal häufig auch noch diverse Zwischenschritte erforderlich sind, auf die wir hier nicht näher eingehen wollen. Zum anderen stellt das von einem einfachen Compiler erzeugte Objektprogramm für den Programmierer meist einen großen schwarzen Kasten dar, in den hineinzusehen ihm verwehrt bleibt. Er kann das Programm in der Regel nicht einfach unterbrechen, um sich bestimmte Variablenwerte anzusehen oder Variablen bestimmte Testwerte zuweisen, um damit dann einen kritischen Programmteil ausführen zu lassen, mal eben eine Zeile ändern und was der Annehmlichkeiten beim Programmtest mit einem Interpreter mehr sind. Bessere Compiler bieten zwar eine Reihe von Optionen und Hilfsprogrammen für die Fehlersuche und das Programmtesten an, jedoch bleibt auch hier, verglichen mit einem Interpreter, diese Arbeit reichlich unbequem. Ideal ist es sicherlich, äquivalente Interpreter und Compiler zur Verfügung zu haben. Auch gewisse Mischformen wie zum Beispiel bei der Sprache Forth sind hier interessant.

Strubs — eine Mischung von Interpreter und Compiler

Um nun aber auf das Programm Strubs zurückzukommen: Auch hier haben wir es in gewisser Hinsicht mit einer Mischform zu tun. Das selbst nicht lauffähige Quellprogramm, wel-

ches der Programmierer unter Benutzung der neuen Befehle erstellt, wird von Strubs nicht in Maschinensprache übersetzt, sondern in ein normales Basic-Programm, das dann wie bisher interpretiert wir. Dabei werden Programmteile, die keine Erweiterungen enthalten, mehr oder weniger unverändert übernommen. Dieses von Strubs erzeugte Objektprogramm kann nun wie jedes andere Basic-Programm — auch mit Hilfe von Toolkits — gelistet, ausgetestet und sogar geändert werden. Schließlich ist es dann noch möglich, dieses Objektprogramm mit Hilfe eines Basic-Compilers, wie zum Beispiel dem Austro Compiler, weiter zu übersetzen. Besonders hilfreich ist es, daß einander entsprechende Programmzeilen im Quellprogramm und im Objektprogramm gleiche Zeilennummern besitzen, so daß der Programmierer sich ohne Schwierigkeiten im Objektraum zurechfinden kann. Gegenüber der Methode, den Basic-Interpreter zu erweitern, bietet dieses Verfahren Geschwindigkeitsvorteile: Diese ergeben sich einerseits aus der Tatsache, daß alle Kommentare und Leerzeichen gelöscht werden können, andererseits wird wie beim Compiler ein Teil der Sucharbeit während der Übersetzung erledigt. Dadurch werden zum Teil erst neue Anweisungen ermöglicht, deren Realisierung im Rahmen einer Interpretererweiterung zu aufwendig wäre.

Schon durch die Suche nach Sprungzielen wirkt der Basic-Interpreter langsam genug: Bei jedem Sprung wird das Programm von Anfang an durchsucht, bis die entsprechende Zeilennummmer gefunden ist. Deshalb empfiehlt es sich auch, häufig aufgerufene Unterprogramme möglichst an den Programmanfang zu setzen, da sie dann schneller gefunden werden.

Wer sucht, der findet: aber wann?

Daß sich die Suchzeit überhaupt in Grenzen hält, liegt nun daran, daß der Programmtext selbst nicht durchsucht werden muß. Vielmehr braucht der Interpreter nur entlang der Kette aus Zeilennummern und Zeigern zur nächsten Zeile zu suchen, bis die gewünschte Zeilennummer gefunden ist (Bild 1). Sollte nun der Interpreter aber bei nicht erfüllter Bedingung in einer IF-Anweisung das zugehörige ELSE suchen, bei nicht erfüllter Eingangsbedingung einer FOR-Schleife das zugehörige NEXT oder zu einem WHILE das ENDWHILE, dann müßte der gesamte Programmtext selbst durchsucht werden.

Interpreter durchlaufen jede Schleife mindestens einmal

Deshalb arbeiten die Basic-Interpreter im allgemeinen so, daß solche Blöcke — wie die FOR-Schleife — mindestens einmal durchlaufen werden. Deshalb muß bei solchen Interpretern — sofern sie überhaupt ein ELSE kennen — dieses in der gleichen Programmzeile wie das dazugehörige IF stehen. Deshalb kennt zum Beispiel Simons Basic die REPEAT-UNTIL-Anweisung, die immer mindestens einmal durchlaufen wird, nicht aber die WHILE-Anweisung. Ein Precompiler aber kann bei der Übersetzung den Abschlußbefehlen eines Blockes — wie ELSE oder ENDWHILE — ihre Zeilennummern zuordnen, so daß beim Programmlauf nicht mehr der Programmtext selbst, sondern nur die Kette der Zeilennummern durchsucht werden muß.

Vorübersetzung nicht nur beim Precompiler

Die Methode der Vorübersetzung zur Erhöhung der Laufgeschwindigkeit benutzt im übrigen auch der Basic-Interpreter des Commodore 64. Und zwar findet sich die Übersetzungsfunktion im Editor: Sofort bei der Eingabe einer Zeile werden die Basic-Befehle, die aus mehreren Zeichen bestehen, in nur 1 Byte lange Zeichen, die sogenannten TOKEN, übersetzt. Eine Liste dieser Token findet sich zum Beispiel im Programmierhandbuch zum VC 20. Diese Vorübersetzung bringt zwar einen schönen Gewinn an Geschwindigkeit, hat allerdings den Nachteil, daß Programmtexte nicht mehr mit komfortableren Editor- beziehungsweise Textprogrammen erstellt werden können. Für uns ist jedoch vor allen Dingen wichtig, daß diese Token berücksichtigt werden müssen, falls der Befehlsvorrat von Strubs erweitert werden soll, oder falls Programme für Interpretererweiterungen wie Simons Basic bearbeitet werden sollen. Aber auf diesen Punkt werden wir ausführlicher eingehen.

Wenn wir mit Strubs arbeiten, haben wir es — wie bei jedem Compiler — mit (mindestens) drei Programmen zu tun: Dem Übersetzungsprogramm, dem Quellcode (Quellprogramm) und dem lauffähigen Objektprogramm. Diese Programme müssen sich nun irgendwie den zur Verfügung stehenden Speicherplatz teilen. Daß das Übersetzungsprogramm, um arbeiten zu können, im Hauptspeicher stehen muß, versteht sich von selbst.

Eine Möglichkeit wäre nun, daß das Übersetzungsprogamm das Quellprogramm von der Diskette einliest, und gleichzeitig das erzeugte Objektprogramm auf Diskette schreibt. Der Compiler zu Simons Basic arbeitet zum Beispiel nach dieser Methode. Da ein Compiler jedoch einen Programmtext in der Regel mindestens zweimal durchliest — man spricht in diesem Fall von 2-Pass-Compilern —, ist es günstiger, wenn das Quellprogramm sich ebenfalls im Hauptspeicher befindet. Diesen Weg gehen zum Beispiel Pascal 64 und Strubs. Um nun den zur Verfügung stehenden Platz aufzuteilen, benutzt zum Beispiel Pascal 64 eine sehr einfache und wirksame Methode:

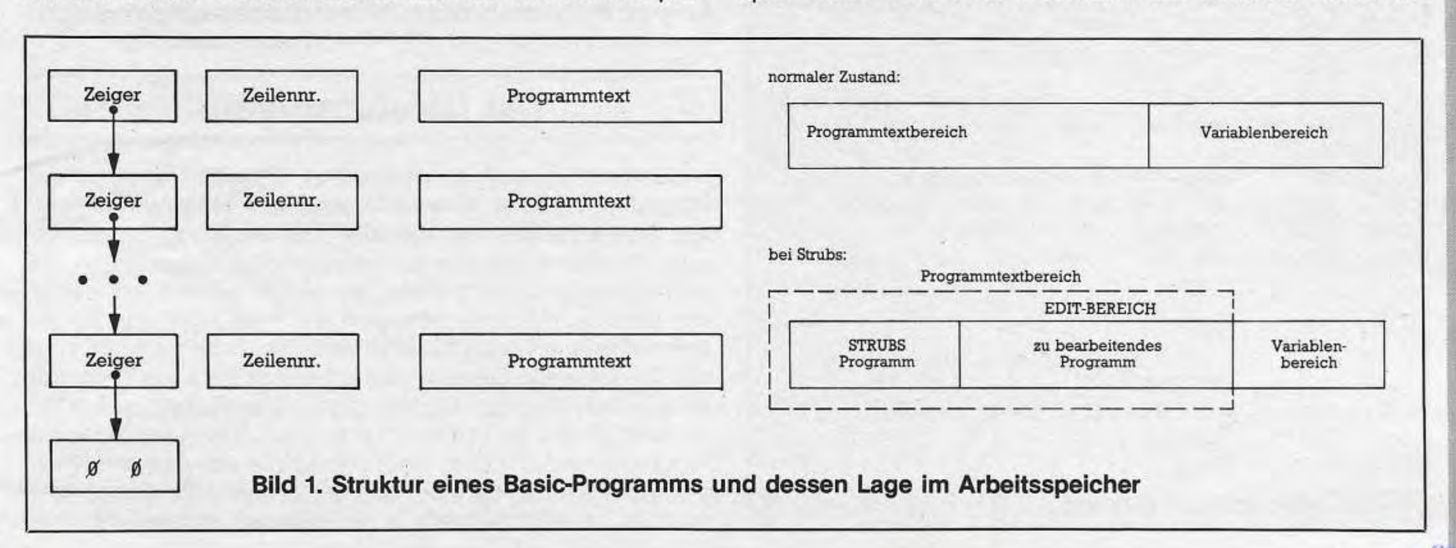


Bild 1. Struktur eines Basic-Programms und dessen Lage im Arbeitsspeicher

Der Compiler ist selbst in Basic geschrieben und enthält eine unsichtbare Zeile mit der Zeilennummer 0, die ihrerseits einen Sprung zum Übersetzungsprogramm enthält. Ein Pascalprogramm wird nun einfach mit den Zeilennummern zwischen 1 und 9999 in das Compilerprogramm eingefügt.

Dieses Verfahren hat allerdings den Nachteil, daß Programme immer nur zusammen mit dem Compiler abgespeichert und editiert werden können. Insbesondere ist es damit nicht möglich, Quellprogramme aus fertigen Bausteinen (Modulen) zusammenzusetzen.

Strubs geht andere Wege

Aus diesem Grund wurde für Strubs ein anderer Weg gewählt: Entsprechend Bild 1 wurde der Speicher des Commodore 64 in drei Bereiche aufgeteilt. Am Anfang des Arbeitsspeichers steht das Programm Strubs (Pointer in Zelle 43/44). Der Edit-Bereich für Quellprogramme beginnt bei (Wert der Variablen EA). Daran anschließend befindet sich der (gemeinsame) Variablenbereich (Pointer in Zelle 45/46). Um nun vom Edit-Bereich aus bequem in den anderen Speicherbereich umschalten zu können, benutzt Strubs selbst eine kleine Interpretererweiterung, die, wie versprochen, kurz vorgestellt werden soll.

Die Eingabe von »!« bewirkt nun dasselbe wie die Befehlsfolge »POKE 44,8: RUN«. Das entsprechende Assemblerlisting findet sich in Bild 2. Das kleine Programm »Erweiterung« holt zunächst den nächsten Befehl. Dann muß für die Routine »Befehl ausführen« der Status gerettet werden, da die CHARGET-Routine damit wichtige Informationen übermittelt. (dies ist wichtig und wurde in dem unten erwähnten Buch übersehen). Nachdem verglichen wurde, ob ein neuer Befehl vorliegt, wird dann entsprechend zum normalen Programmverlauf oder zur Erweiterungsroutine verzweigt. Für eigene Versuche mit Interpretererweiterungen können an dieser Stelle beliebige Maschinenprogramme (gegebenenfalls mit weiteren Decodierungen) gesetzt werden. Nur sollte zum Abschluß — anders als hier, wo ein Basic-Befehl aufgerufen wird — ein Sprung zur Interpreterroutine $A7E4 erfolgen, wo dann der nächste Befehl bearbeitet wird. Um nun die Erweiterung in den Basic-Interpreter einzubinden, benötigen wir dann nur noch eine kurze Initialisierungsroutine, die den Zeiger in $0308 auf den Anfang der Erweiterung setzt.

Wer selbst solche Erweiterungen entwickeln möchte, findet weitere Informationen und viele Anregungen in dem Buch »64 Intern« von Data Becker. Für weitergehend Interessierte empfiehlt sich die gut verständliche Einführung »Compilerbau« von N. Wirth, Teubner, Stuttgart 1981.

```
Erweiterung:
02C0  207300   JSR 0073   ; Charget, nächstes Zeichen holen
02C3  08       PHP        ; Status retten
02C4  C921     CMP #21    ; »!«, neuer Befehl?
02C6  F004     BEQ 02CC
02C8  28       PLP        ; nein, dann Status wiederherstellen
02C9  4CE7A7   JMP A7E7   ; und normalen Befehl ausführen
02CC  28       PLP        ;
02CD  A908     LDA #08    ; Erweiterungsroutine:
02CF  852C     STA 2C     ; entspricht Poke 44,8: RUN
02D1  A98A     LDA #8A    ; RUN-TOKEN
02D3  4CE7A7   IMP A7E7   ; Befehl ausführen

INIT:
02EE  A9C0     LDA #C0    ; Erweiterung, Low Byte
02F0  8D0803   STA 0308
02F3  A902     LDA #02    ; Erweiterung, High Byte
02F5  8D0903   STA 0309
02F8  60       RTS
```

Bild 2. Interpreter-Erweiterung

Anschließend wollen wir nun das Programm Strubs vorstellen. Am Anfang der Programmentwicklung standen folgende Vorstellungen, die durch das Programm erfüllt werden sollten:

1. Unabhängigkeit von Zeilennummern
2. Unterstützung strukturierter Programmierung
3. Unterstützung modularer Programmentwicklung
4. Erweiterung der Dokumentationsfähigkeit des Programmtestes. Dabei sollte das Programm
5. einfache Handhabung gewährleisten und
6. effiziente Fehlersuche ermöglichen.

Die Unabhängigkeit von Zeilennummern wird erreicht durch die Verwendung beliebig langer Labels oder relativer Sprünge anstelle von Zeilennummern.

Strubs stellt sich vor

Die wichtigsten Kontrollstrukturen höherer Programmiersprachen werden von Strubs zur Verfügung gestellt:

IF — THEN — FI
IF — THEN — ELSE — FI
WHILE — EWHILE
REPEAT — UNTIL
LOOP — EXIT (beliebig oft) — ELOOP
CASEOF — OF (beliebig oft) — ELSE (optional) — ECASE

Durch die Unabhängigkeit von Zeilennummern und eine EXTERN-Deklaration wird das Anlegen einer Modulbibliothek — sowohl auf Quellprogramm- als auch auf Objektprogrammebene — unterstützt.

Mit Strubs werden Sie ein vielseitiges Werkzeug in Händen halten

Der Dokumentationsfähigkeit des Programmtestes dienen neben den bereits erwähnten Marken und Kontrollstrukturen ein Tabulator und Kommentare an beliebiger Stelle auch innerhalb eines Variablennamens (zum Beispiel A'US'G'ABE'% = AG%).

Programmtests können wie gewohnt mit dem normalen Basic-Editor geschrieben werden.

Schließlich werden wir zur Illustration der Erweiterung des Befehlssatzes von Strubs noch eine MAKRO-Funktion implementieren. Von besonderer Bedeutung ist, daß das Programm von Anfang an unter dem Aspekt möglichst einfacher Erweiterbarkeit konzipiert wurde. Damit konnte das Programm im Bootstrapping-Verfahren entwickelt werden, so daß es jetzt selbst sowohl als Quellprogramm wie auch als Objektprogramm vorliegt. Wem es Spaß macht, der mag Strubs einfach auch als ein generelles Übersetzungsprogramm zur Aufbereitung von Programmtexten auffassen und seine gegenwärtigen Funktionen als Beispiel möglicher Implementationen.

Die Befehlsstruktur

Gehören Sie auch zu denjenigen, die sich manchmal ein Programm aus einer Zeitschrift vornehmen, um zu analysieren, wie es arbeitet oder um eventuell Teile des Programms für eigene Programmprojekte zu verwenden? Dann erinnern Sie sich bestimmt an Programme, bei denen Sie sich verzweifelt von Sprung zu Sprung bewegen und nach nicht allzu langer Zeit vollkommen den Überblick verlieren. Oder vielleicht kennen Sie folgende Situation: Sie schreiben ein Programm und erinnern sich angesichts eines bestimmten Problems, daß Sie ein ganz ähnliches Problem schon einmal in einem anderen Programm gelöst haben. Aber sobald Sie sich den alten Programmtext vornehmen, um den entsprechenden Programmteil in ihr neues Programm zu übernehmen, müssen Sie ent-

täuscht feststellen, daß diese spezielle Problemlösung so sehr in das Programmgeflecht verwoben ist, daß es Ihnen weitaus einfacher scheint, den entsprechenden Programmteil vollkommen neu zu entwickeln.

Die Ursache für solche Erscheinungen liegt zum Teil darin, daß viele Basic-Programme mehr oder weniger aus der Sicht des Computer der »Basic-Maschine« — direkt am Computer nach dem Verfahren von Versuch und Irrtum entwickelt werden. Das kann in Einzelfällen sogar soweit führen, daß man zum Schluß zwar sieht, daß das Programm läuft, aber selbst nicht so recht weiß, warum eigentlich, und wie es funktioniert. Der Hauptgrund für solche Unübersichtlichkeit aber liegt in der Verwendung zahlreicher wilder Sprünge und ausgefallener Programmier-Tricks. (Daß die Verwendung von GOTO-Anweisungen den mathematischen Beweis für die Korrektheit von Programmen praktisch unmöglich macht, ist für den Informatiker interessant, braucht uns hier aber nicht zu interessieren.)

Den entgegengesetzten Weg geht die strukturierte Programmierung. Sie bedeutet vor allem sorgfältige Planung und den Verzicht auf GOTOs und unübersichtliche Programmiertricks. Hier steht die systematische Analyse des Problems im Vordergrund. Die eigentliche Codierung, das heißt die Formulierung des Programmtextes in einer bestimmten Programmiersprache, spielt nur eine untergeordnete Rolle.

In der Problemanalyse geht es darum, ein gegebenes Problem in relativ selbständige Teilprobleme zu zerlegen und deren Beziehungen zueinander festzulegen. Den Aufbau des Programms Strubs mit den jeweiligen Zeilennummern können Sie Bild 3 entnehmen. Das komplette Objektprogramm ist ebenfalls abgedruckt (siehe Listing 1).

Entsprechend setzt sich das strukturierte Programm aus einer Reihe möglichst selbständiger Programmeinheiten zusammen. Dieses Vorgehen spiegelt sich im Konzept der Blöcke und Module.

Ein Block ist eine Anweisung oder eine Folge von Anweisungen mit genau einem Eingang und genau einem Ausgang. Das heißt man darf weder in einen solchen Block hineinspringen, noch aus diesem Block herausspringen. Solche Blöcke können entweder aneinandergereiht oder beliebig tief ineinandergeschachtelt werden; sie dürfen sich aber nicht überschneiden. In letzterer Hinsicht verhält es sich mit diesen Blöcken also genauso, wie bei den bekannten FOR-Schleifen in Basic.

Ein strukturiertes Programm besteht nun ausschließlich aus einer geordneten Hierarchie solcher Blöcke. Der kleinste mögliche Block besteht aus einer einzelnen Anweisung, wie zum Beispiel PRINT "Text". Der größte, umfassendste Block besteht aus dem Programm selbst.

Da ist zunächst einmal die einfache IF-Anweisung, die schon von Basic her bekannt ist. Dieses normale Basic-IF kann natürlich, wie alle Basic-Befehle, weiterhin benutzt werden. Zusätzlich bietet Strubs aber eine erweiterte Form, bei welcher der THEN-Teil nicht auf den Rest einer Programmzeile begrenzt ist, sondern beliebig viele Zeilen umfassen kann, die durch den Befehl '!FI' — einfach ein umgedrehtes IF — abgeschlossen werden. Ein Beispiel:

```
10 ! IF X=Y THEN
20 :   PRINT "X und Y"
30 :   PRINT " SIND GLEICH"
...
99 !FI
```

Ist die Bedingung hinter IF erfüllt, so werden die Zeilen zwischen der IF- und der FI-Anweisung ausgeführt, ansonsten wird das Programm sofort hinter der FI-Zeile fortgesetzt.

Daneben existiert selbstverständlich auch die vollständige Form

```
10 !IF X=Y THEN
20:   PRINT "GLEICH'
...
50 !ELSE
60 :   PRINT "UNGLEICH"
...
99 !FI
```

Ist die Bedingung erfüllt, dann wird der Block zwischen IF und ELSE ausgeführt, sonst der Block zwischen ELSE und FI.

Für den Fall, daß mehr als nur zwei Fälle zu unterscheiden sind, bietet Strubs die CASE-Anweisung:

```
10 !CASEOF X<0 THEN
15 :      PRINT "KLEINER ALS 0"
...
40 ! OF X=0 THEN
45 :      PRINT "GLEICH 0"
...
60 ! OF X>0 AND Y <OTHEN
65 : PRINT "X>0 UND Y<"
...
80 ! ELSE
85 :      PRINT "KEINER DER FÄLLE TRIFFT ZU"
99 ! ECASE
```

Mit dieser Struktur können beliebig viele Fälle unterschieden werden, wobei jedes OF mit einer beliebigen Bedingung verbunden werden kann. Es sollte aber darauf geachtet werden, daß sich die Bedingungen gegenseitig ausschließen (sonst wird das erste Auftreten einer erfüllten Bedingung gewählt). Nach der Bearbeitung des entsprechenden Falles wird das Programm immer hinter ECASE fortgesetzt. Die Möglichkeit, daß keiner der Fälle zutrifft, kann mit Hilfe der ELSE-Anweisung behandelt werden. Ist dies nicht erforderlich, kann der ELSE-Teil auch entfallen.

Damit kommen wir nun zu den Schleifen. Die FOR-Schleife kann wie bisher benutzt werden. Die WHILE-Schleife wird durchlaufen, solange die Bedingung erfüllt ist. Anschließend wird das Programm hinter EWHILE fortgesetzt. Da die Bedingung am Anfang der Schleife abgefragt wird, kann es vorkommen, daß die Schleife auch überhaupt nicht durchlaufen wird.

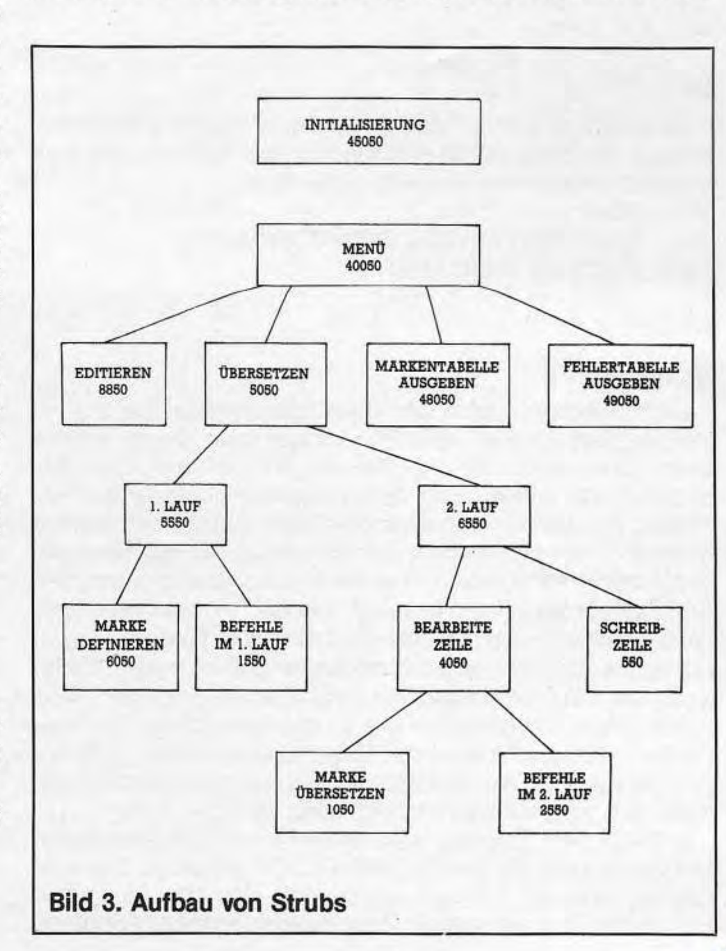


Bild 3. Aufbau von Strubs

```
510 '********************************
520 '* GESCHACHTELTE LOOP-BLOECKE    *
530 '********************************
540 '
620 !LOOP 'L1
630 :   PRINT"AEUSSERE LOOP1"
640 :   IF X=1 THEN !EXIT 'LOOP1
650 :   !LOOP 'L2
660 :      PRINT "INNERE LOOP2"
670 :      IF X=0 THEN !EXIT 'LOOP2
680 :   !ELOOP ' L2
690 '   HIER WIRD PROGR. NACH EXIT LOOP2 FORTGESETZT
700 :   X=X+1
710 :   !IF X=2 THEN
720 :       PRINT "LOOP 1 VERLASSEN:":!EXIT 'LOOP1
730 :   !FI
740 :   X=X+1
750 !ELOOP 'L1
760 PRINT"HIER WIRD PROGRAMM NACH EXIT LOOP 1 FORTGESETZT"
770 '
READY.
```

Bild 4. Geschachtelte Loop-Schleife

Ein Beispiel:

```
10 ! WHILE X<5 !DO
20 :  PRINT "IMMER NOCH KLEINER ALS 5"
30 : X = X + 1
...
99 !EWHILE
```

Von der WHILE-Schleife unterscheidet sich die REPEAT-Schleife in zwei Punkten: Erstens wird die Schleife durchlaufen, bis die Bedingung erfüllt ist, also solange sie nicht erfüllt ist. Zweitens wird die Bedingung erst am Ende der Schleife abgefragt, so daß die Schleife immer mindestens einmal durchlaufen wird. In diesem wie im nächsten Beispiel bezieht sich die Zeile 30 auf den Fall, daß X beim Eintritt in die Schleife größer als 5 ist:

```
10 ! REPEAT
20 : PRINT "X KLEINER ALS 5"
30 : PRINT "VIELLEICHT ABER AUCH NICHT"
40 : X = X+1
...
99 ! UNTIL X >= 5
```

Eine weniger weit verbreitete, aber sehr mächtige Schleifenstruktur stellt die LOOP-Schleife dar (sie befindet sich zum Beispiel in der Programmiersprache ADA):

```
10 ! LOOP
30 :  PRINT "EVENTUELL GROESSER ALS 5"
40 :  IF X>=5 THEN !EXIT
50 :  PRINT "KLEINER ALS 5"
60 :  X = X+1
...
99 !ELOOP
```

Es handelt sich dabei um eine Endlosschleife, welche mit Hilfe des Befehls EXIT verlassen werden kann. Diese Schleife bietet im wesentlichen zwei Vorteile: Zum einen muß die Bedingung nicht entweder am Anfang oder am Ende der Schleife stehen, sondern kann an jeder beliebigen Stelle innerhalb des Blockes abgefragt werden. Darüber hinaus ist das Beenden der Schleife nicht nur von einer Bedingung abhängig, sondern die LOOP-Schleife kann beliebig viele EXIT-Anweisungen enthalten (dadurch wird nicht die oben erwähnte Forderung nach nur einem Ausgang verletzt, da das Programm in allen Fällen hinter dem ELOOP fortgesetzt wird). Damit eignet sich diese Konstruktion insbesondere gut für die Behandlung von Ausnahmen wie zum Beispiel von Eingabebefehlen etc. (eine Angelegenheit, die zum Beispiel in Pascal recht umständlich sein kann, falls man auf GOTOs verzichten will oder muß).

In Bild 4 (das Zeichen ' kennzeichnet Kommentare) sehen Sie ein Beispiel für geschachtelte LOOP-Schleifen. Die Ausführung einer EXIT-Anweisung bewirkt die Fortsetzung des Programms bei der ersten Zeile hinter derjenigen Schleife, welche diese EXIT-Anweisung am nächsten umschließt. Im Beispiel enthält die äußere Schleife zwei EXIT-Anweisungen — eine davon vor, die andere hinter der inneren Schleife. Die innere Schleife enthält eine EXIT-Anweisung. Grafisch lassen sich blockstrukturierte Programme am besten durch Struktogramme — anstelle der verbreiteten Flußdiagramme — darstellen. Das Struktogramm für die LOOP-Schleifen finden Sie in Bild 5. Über die Diagramme der anderen Strukturen und den Umgang mit Struktogrammen können Sie sich an anderer Stelle in dieser Zeitschrift oder in den unten aufgeführten Büchern informieren. Kommen wir nun zu den Modulen. Dabei handelt es sich um besondere Blöcke, die ein bestimmtes Teilproblem — beispielsweise das Zeichnen einer Linie in einem Grafikprogramm — unter möglichst weitgehender Unabhängigkeit vom restlichen Programmtext bearbeiten. Stellen Sie sich vor, Sie finden in einer Zeitschrift ein Pascal-Programm zur Einstellung von Grafiken. Dieses Programm benutzte zum Beispiel die Anweisung PLOT (X,Y) zum Zeichnen eines Punktes mit den Koordinaten X und Y. Ihr Freund möge eine Sprache wie Super-Pascal besitzen, die diese Anweisung standardmäßig enthält. Er tippt das Programm ein, es läuft — fertig. Sie selbst besitzen aber nur ein mageres Mini-Pascal, das diesen Befehl nicht kennt. Nun, mit Pascal ist das kein Problem: Sie schreiben sich eine Procedur PLOT (X,Y), fügen diese in das Programm ein — fertig. An dem Programmtext selbst brauchen Sie nicht die geringste Änderung vorzunehmen. Ja, brauchen ihn nicht einmal näher anzusehen. Woran liegt das?

Vom Problem her — dem Erstellen einer Grafik — ist das Zeichnen eines Punktes das Zeichnen eines Punktes. Das einzige, was interessiert, ist, daß dazu zwei Koordinaten erforderlich sind. Dieser Tatsache trägt die Sprache Pascal dadurch Rechnung, daß sie keinen Unterschied macht zwischen dem Aufruf von vorgegebenen Standardanweisungen und selbstdefinierten Prozeduren.

Wenn Sie in einem Basic-Programm irgendwo eine Zeile PRINT "TEXT"stehen haben, erwarten Sie selbstverständlich, daß dadurch nicht 50 Zeilen weiter der Wert der Variablen A verändert wird. Entsprechend sorgt nun Pascal dafür, daß eine selbstdefinierte Prozedur genausowenig Auswirkungen auf andere Programmteile hat wie der Aufruf einer Standard-Anweisung. Die interne Arbeitsweise einer solchen Prozedur wird vor der Programmumgebung genauso versteckt, wie dies bei der internen Arbeitsweise von im Sprachumfang enthaltenen Anweisungen der Fall ist. Entsprechend nennt man dieses Konzept auch »Information Hiding«. Programmiersprachen wie ADA, MODULA oder SIMULA bieten in dieser Hinsicht noch sehr viel weitergehende Möglichkeiten als Pascal.

Schnittstellen:

Der Datenaustausch mit der Umgebung eines Moduls erfolgt über genau definierte Schnittstellen. Bei einer solchen Schnittstelle handelt es sich um eine Menge derjenigen Annahmen, die die Programmumgebung über ein Modul macht — das heißt welche Daten es als Eingabe erwartet, welche Daten es daraufhin wieder ausgibt und welche anderen Module es seinerseits benötigt.

Modulbibliothek:

Die relative Eigenständigkeit solcher Module sorgt nun nicht nur für einfache Änderbarkeit und Erweiterbarkeit, sondern ermöglicht auch das Anlegen einer sogenannten Modulbibliothek. Eine solche Bibliothek enthält eine Reihe von Programmbausteinen, die je nach Bedarf in zu entwickelnde Programme eingefügt werden können. Dabei kann es sich um Sortierroutinen, Grafik-Routinen, mathematische und statistische Routinen und so weiter handeln. Aber auch die Entwicklung von Spielen läßt sich auf diese Weise vereinfachen: Man kann Bibliotheken fertiger Sprites, von eigenen Zeichensätzen oder von diversen Soundroutinen anlegen.

Das wichtigste Hilfsmittel zur Unterstützung modularer Programmentwicklung stellen sicherlich die lokalen Variablen dar.

Leider gibt es solche nicht in Basic und auch Strubs kann keine lokalen Variablen bieten. So ist es auch weiterhin erforderlich, beim Einsetzen oder Ändern eines Moduls darauf zu achten, ob und an welchen Stellen Variablen des Moduls in anderen Programmteilen benutzt werden, und gegebenenfalls Umbenennungen vorzunehmen. Der zweite große Nachteil von Basic — die leidigen Zeilennummern — braucht uns dagegen nur noch wenig zu beschäftigen. Strubs bietet alle Möglichkeiten, die erforderlich sind, um ein Programm vollkommen unabhängig von Zeilennummern zu schreiben. Als erstes sind da natürlich die oben besprochenen Kontrollstrukturen zu nennen.

Darüber hinaus können bei allen Sprüngen Zeilennummern durch Label (Marken) ersetzt werden. Solche Label werden durch das Zeichen »£« gekennzeichnet und abgeschlossen durch ein Leerzeichen, Doppelpunkt, Komma oder Zeilenende. Die dürfen zwar reservierte Basic-Worte enthalten, dann können sich aber wegen der vorhin schon erwähnten Token bei der Ausgabe der Markentabelle seltsame Effekte ergeben. Die Label werden definiert, indem sie an den Anfang einer Zeile gesetzt werden und können beliebig lang sein:

```
10  £X-AUSGEBEN:
20 :  PRINT "X:";X
30  RETURN
...
200 X=1:GOSUB £X-AUSGEBEN
210 X=2:GOSUB £X-AUSGEBEN
```

Schließlich bietet Strubs noch die Möglichkeit relativer Sprünge. Diese dienen vor allem dazu, kurze Schleifen innerhalb einer einzigen Zeile zu konstruieren, ohne dafür extra ein Label zu definieren:

```
90 NC=NC+1:C=PEEK(NC):IF C>0 THEN Z$+CHR$(C):
GOTO £THIS
```

Der Befehl GOTO £THIS bewirkt einen Sprung an den Anfang derjenigen Zeile, in der dieser Befehl steht.

Da bei der Arbeit mit Strubs Quellprogramme in der Regel weit umfangreicher als die Objektprogramme sind, bietet Strubs die EXTERN-DEKLARATION, die es ermöglicht, Module und Programmteile getrennt zu übersetzen und erst auf der Objektprogrammebene zusammenzufügen. Hierbei müssen die einzelnen Programmteile allerdings verschiedene Zeilennummern belegen. In der Extern-Deklaration wird ein Name vereinbart, unter dem ein Programm ein externes Modul ansprechen kann. Diesen Namen wird die Einsprungadresse (bei Maschinenprogrammen) beziehungsweise die Zeilennummer bei Basic-Routinen zugewiesen:

```
20  REM VEREINBARUNG:
30 ! EXT: £MAPRO:740,£PLOT: 50000
...
90 REM AUFRUF:
99 SYS £MAPRO: X=13:Y=90:GOSUB £PLOT
```

Kommen wir zur Dokumentation: Vom Hobby-Programmierer kann kein Mensch erwarten, daß er Berge von Dokumentationsmaterial anlegt, die den Umfang des Programmtextes um ein Vielfaches übersteigen. Deshalb ist es gerade hier wichtig, Programme weitgehend selbstdokumentierend zu schreiben. Im Gegensatz zu höheren Programmiersprachen mit ihren zahlreichen Deklarationspflichten ist der Basic-Programmierer nahezu ausschließlich auf Kommentare angewiesen. Da Strubs Kommentare bei der Übersetzung eliminiert, stehlen diese weder Speicherplatz noch Laufzeit. Der Programmierer kann also ohne Bedenken einen exzessiven Gebrauch von Kommentaren machen.

Kommentare werden gekennzeichnet durch das Zeichen » ' «. Steht dieses Zeichen direkt am Zeilenanfang, so wird die ganze Zeile gelöscht. Sonst wird der Programmtext bis zum zweiten » ' « oder bis zum Zeilenende überlesen. Außer innerhalb von Befehls- und Markennamen können Kommentare an jeder beliebigen Programmstelle eingefügt werden. Kommentare, die in das Objektprogramm übernommen werden sollen, können wie bisher mit REM in den Programmtext eingefügt werden. Beispiel:

```
10 ' DIESE ZEILE WIRD VOLLSTÄNDIG GELÖSCHT
20 A a'US a'G a'ABE a'$="ENTSPRICHT AG$"
   'KOMMENTAR
```

Die Lesbarkeit von strukturierten Programmen wird verbessert durch das Einrücken von Zeilen entsprechend der Blockstruktur. Hierzu dient der Tabulator (Bild 4): Ein Doppelpunkt am Zeilenanfang gefolgt von Leerzeichen.

Wenn Sie Quellprogramme schreiben, achten Sie vor allem auf folgendes: Jeder der neuen Befehle muß durch ein Ausrufezeichen gekennzeichnet werden, zum Beispiel !REPEAT, und benötigt — außer !DO — eine eigene Programmzeile. Folgende Konstruktion ist also zum Beispiel nicht erlaubt:

```
40 !REPEAT X=X+1 !UNTIL X>5
```

Marken beginnen grundsätzlich mit einem Pfund-Zeichen £.

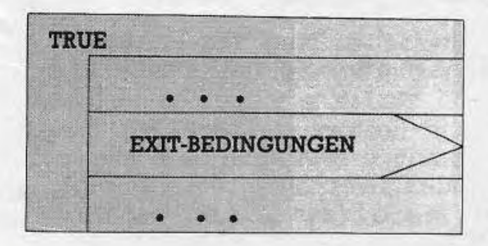


Bild 5. Struktogramm der Loop-Schleife

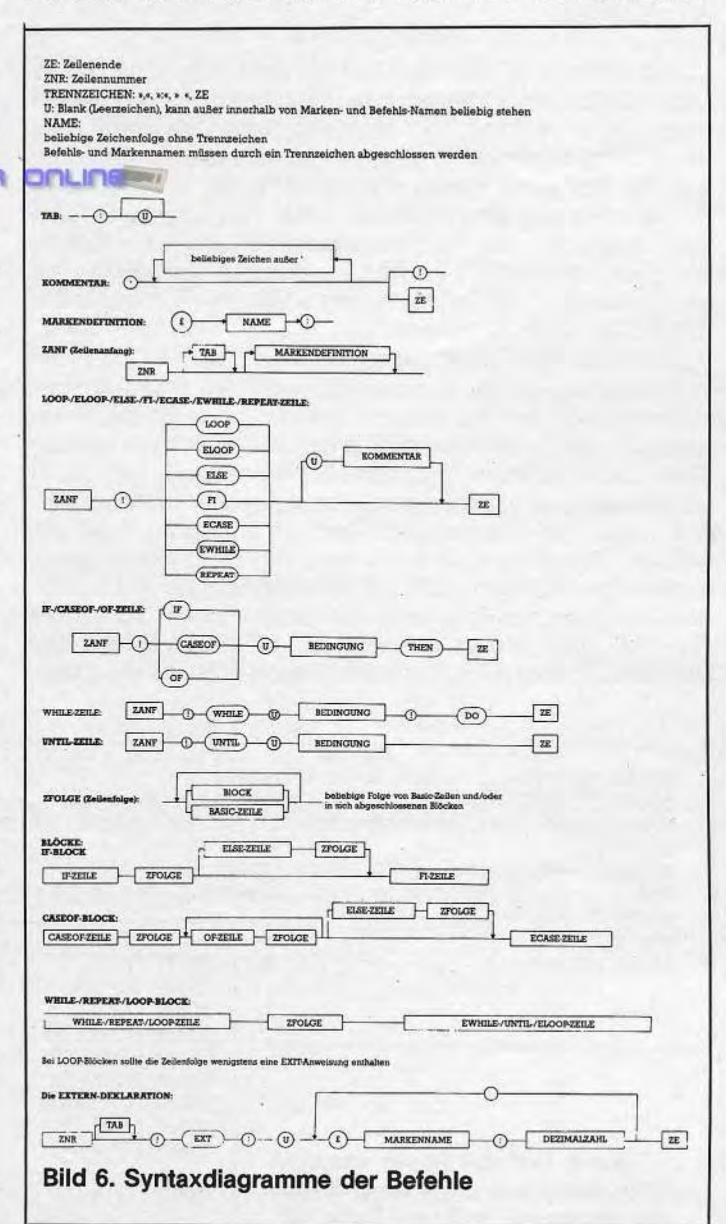


Bild 6. Syntaxdiagramme der Befehle

Um die Korrektheit von Konstruktionen zu überprüfen, können Sie die Syntax-Diagramme in Bild 6 benutzen. Wenn sich ein Weg entlang der Linien finden läßt, der der Konstruktion entspricht, dann ist diese in Ordnung. Sie können sich aber auch an den verschiedenen Beispielen im Rahmen dieses Artikels orientieren.

Das beste Beispiel für ein Quellprogramm erhalten Sie, wenn Sie die Programmdiskette mit dem Programm Strubs bestellen. Diese Diskette enthält neben dem lauffähigen Objektprogramm auch das ausführlich dokumentierte Quellprogramm von Strubs, dessen Abdruck aus Platzgründen nicht möglich ist.

Programmentwicklung mit Strubs

Bei der Blockschachtelung ist darauf zu achten, daß sich verschiedene Blöcke nicht überschneiden dürfen und daß jeder Block korrekt abgeschlossen wird. Hierbei kann man sich immer das Beispiel der FOR-NEXT-Schleifen in Basic vor Augen halten. Vollkommen unmöglich ist beispielsweise folgende Konstruktion:

```
10 !REPEAT
20 : !WHILE ... !DO
30 !UNTIL ...
40 : !EWHILE
```

Nun wird es aber allmählich Zeit, mit der Praxis zu beginnen. Laden Sie das Programm Strubs in Ihren Computer und starten es mit »RUN«. Nun erscheint ein Menü. Geben Sie hier »E« ein, um in den Editbereich zu gelangen (siehe dazu die erste Folge). Der Computer meldet sich mit »READY«, das heißt also, Sie befinden sich jetzt im Direktmodus. Hier können Sie nun (fast) so arbeiten, als sei Strubs gar nicht vorhanden. Geben Sie zunächst »NEW« ein. Jetzt können Sie das kleine Programm aus Bild 7 eintippen und wie sonst gewohnt mit »SAVE "RENUMBER.QP",8« abspeichern.

Mit diesem »QP« hat es folgende Bewandtnis: Bei Compilern ist es allgemein üblich, die verschiedenen Files, die zu den einzelnen Phasen der Übersetzung gehören, einheitlich zu kennzeichnen, der Austro-Compiler, arbeitet zum Beispiel mit den Files »name«, »p/name«, »z/name« und »c/name«. Um Quellprogramme und Objektprogramme auseinanderhalten zu können, sollten Sie sich entsprechend von Anfang an daran gewöhnen, dem Programm immer ein »QP« für Quellprogramm beziehungsweise ein »OP« für Objektprogramm hinzuzufügen. Nun kann das Programm übersetzt werden. Geben Sie ein »!RETURN« und es erscheint das Menü von Strubs. Die Übersetzung wird mit »U« gewählt. Strubs fragt nun nach dem Namen für das Objektprogramm. Geben Sie ein: RENUMBER.OP. Da das übersetzte Programm direkt auf Diskette geschrieben wird, achten Sie darauf, daß die Floppy eingeschaltet ist. Nun erscheint auf dem Schirm die Meldung »1. Lauf«, gefolgt von der Ausgabe der Blockstruktur. Nach Beendigung des 2. Laufs sollte die Meldung »0 FEHLER« erscheinen. Ist dies der Fall, dann können Sie mit »E« wieder in den Edit-Bereich gelangen. Hier steht immer noch das Quellprogramm. Um sich das übersetzte Programm anzusehen, laden Sie es mit »LOAD "RENUMBER.OP",8«. Es sollte mit dem Listing in Bild 8 übereinstimmen. Aber starten Sie das Programm nicht.

Jetzt übersetzen Sie einmal genauso verschiedene kleine Testprogramme — testen Sie doch einmal die LOOP, REPEAT-, und WHILE-Befehle — und sehen sich die Ergebnisse an. Dabei werden Sie feststellen, daß einige Bedingungen im Objektprogramm in negierter Form erscheinen. Daß Basic keine boolschen Variablen kennt, hat eine wichtige Konsequenz: Beim Test, ob eine Variable ungleich 0 ist, darf man nicht — wie dies normalerweise häufig in Basic formuliert wird — beispielsweise schreiben

IF A THEN ...,

sondern muß bei jeder Bedingung die vollständige Form

IF A< >0 THEN ...

verwenden. Dies liegt daran, daß die Bedingung NOT(A) außer für —1 immer erfüllt ist.

Zweitens kann man sehen, daß in den Objektprogrammen manchmal neue Zeilen auftauchen, die das Quellprogramm nicht enthielt. Strubs generiert solche Zeilen als Sprungziele. Damit immer Platz für solche Zeilen ist, sollte der Abstand der Zeilennummern im Quellprogramm immer mindestens 2 betragen.

Ist bisher alles wie oben beschrieben verlaufen, dann können Sie sich freuen. Sind irgendwelche Fehler aufgetreten, dann vergleichen Sie noch einmal genau das Testprogramm mit dem Listing in Bild 7 und hoffen Sie, daß der Fehler hier liegt. Wenn Sie keine Abweichungen feststellen, dann haben Sie Pech — Sie haben das Programm Strubs falsch eingetippt.

Wie steht es aber mit Fehlern im Quellprogramm? Syntax-Fehler können sich auf drei verschiedene Arten bemerkbar machen: Vor allem Fehler, die nicht mit den neuen Befehlen zusammenhängen, führen wie gewohnt beim Lauf des Objektprogramms zu den bekannten Fehlermeldungen. Fehler in bezug auf die neuen Befehle quittiert Strubs mit Abbruch der Übersetzung, falls eine Fortsetzung nicht sinnvoll erscheint, oder mit Eintragung in eine Fehlerliste und gleichzeitiger Kennzeichnung der fehlerhaften Zeile im Objektprogramm. Die Fehlerliste kann man sich mit »F« ansehen.

Ein Abbruch der Übersetzung mit entsprechender Fehlermeldung am Bildschirm erfolgt vor allem bei Verstößen gegen die Blockstruktur (und bei Speicherplatzproblemen wie Stack-

```
54990 '******** RENUMBER ***************
55000 £RENUMBER: PRINT"
       ***** RENUMBER ******"
55010 ZA=40*256+1:INPUT"  STARTNR.";ZN:INPUT " SCHRITTWEITE";SW
55020 !LOOP
55030 :  IFPEEK(ZA+1)=0THEN !EXIT 'FERTIG
55040 :  H%=ZN/256:POKE ZA+2,ZN-H%*256:POKE ZA+3,H%:ZA=PEEK(ZA)+256*PEEK(ZA+1)
55050 :  ZN=ZN+SW
55060 !ELOOP
55070 RETURN

READY.
```

Bild 7. Das Quellprogramm des RENUMBER-Befehls

```
55000 PRINT"           ***** RENUMBER ******"
55010 ZA=40*256+1:INPUT"  STARTNR.";ZN:INPUT" SCHRITTWEITE";SW
55020 :
55030 IFPEEK(ZA+1)=0THEN55061
55040 H%=ZN/256:POKEZA+2,ZN-H%*256:POKEZA+3,H%:ZA=PEEK(ZA)+256*PEEK(ZA+1)
55050 ZN=ZN+SW
55060 GOTO55020
55061 :
55070 RETURN

READY.
```

Bild 8. Das von Strubs erzeugte Objektprogramm. Wie es in Strubs einzufügen ist, steht auf Seite 107.



Overflow oder Listen voll). Bei Fehlern mit den Blöcken — zum Beispiel vor einem ELSE fehlt das IF oder zu einem WHILE fehlt das EWHILE etc. — gibt es ein Problem, das nicht nur bei Strubs, sondern generell bei allem Compilern auftaucht: Der Fehler wird nicht an der Stelle seines Auftretens bemerkt, sondern erst viel später. Die Zeilennummer bei einer Fehlermeldung wie »BLOCKSCHACHTELUNG: ANFANG FEHLT«, sagt also nichts weiter aus, als daß der Fehler erst hier bemerkt wurde. Um bei der Suche nach dem Fehler zu helfen, gibt Strubs aber während der Übersetzung ein Schema der Blockstruktur aus, mit dessen Hilfe sich solche Fehler leicht lokalisieren lassen. Bei Meldungen wie »zu viele Marken«, »zu viele WHILE/REPEAT« etc. empfiehlt es sich, das Programm in kleinere Teile zu zerlegen, diese getrennt zu übersetzen und anschließend wieder zusammenzufügen.

Wie man dazu vorgeht, behandeln wir weiter unten. Die entsprechenden Listen sind allerdings so großzügig dimensioniert, daß dieser Fall sehr selten eintreten wird.

Sollte während der Übersetzung aus irgendeinem Grund ein unkontrollierter Programmabbruch erfolgen (zum Beispiel OUT OF MEMORY ERROR), dann empfiehlt es sich mit »GOTO 5000« dafür zu sorgen, daß offene Disk-Files ordnungsgemäß geschlossen werden.

Die Beseitigung von Fehlern, die Strubs bei der Übersetzung entdeckt, gestaltet sich relativ einfach: Notieren Sie sich die Zeilennummern zu jedem Fehler und schalten in den Editbereich (mit »E«). Dort kann das Quellprogramm geändert werden, dann wird mit »!« und Wahl von »U« neu übersetzt. Da das Quellprogramm so lange im Edit-Bereich bleibt, bis es durch »NEW« oder Laden eines anderen Programms gelöscht wird, kann dieser Vorgang so lange wiederholt werden, bis der letzte Fehler beseitigt ist. Sobald die Übersetzung mit der Meldung »0 Fehler« beendet wird, geht es ans Testen des Objektprogramms.

Hierzu wird Strubs durch Eingabe von »S« verlassen. Dadurch wird ein Kaltstart ausgeführt, der die Interpretererweiterung abschaltet und den Speicher säubert. Nun laden Sie das Objektprogramm unter dem Namen, den Sie bei der Übersetzung angegeben haben, und starten es mit RUN. Dieses Programm wird nun wie jedes normale Basic-Programm ausgetestet. Dazu können selbstverständlich auch Toolkits mit TRACE- und DUMP-Funktionen verwendet werden. Da die Zeilennummern denen des Quellprogramms entsprechen, empfiehlt es sich, ein Listing des Quellprogramms zur Hand zu haben. Fehler in der Programmlogik lassen sich damit leichter finden und beheben.

Die Bequemlichkeit, die Strubs dadurch bietet, daß Programmänderungen und Verbesserungen im Objektprogramm selbst vorgenommen und sofort ausgetestet werden können, erfordert auf der anderen Seite allerdings eine gewisse Disziplin, damit die Verbindung zum Quellprogramm nicht verloren geht. Jede vorgenommene Änderung sollte sorgfältig notiert und nicht zu viele Änderungen auf einmal vorgenommen werden. Dann wird wieder das Programm Strubs und das Quellprogramm (in den Editierbereich) geladen. Verbessern Sie das Quellprogramm entsprechend Ihren Notizen und übersetzen es erneut. Dieser Vorgang wird so lange wiederholt, bis das Ergebnis zufriedenstellend ist.

Dieser soeben beschriebene Ablauf kann allerdings in den meisten Fällen vereinfacht werden: Bis auf zwei Ausnahmen können Objektprogramme auch direkt im Editbereich getestet werden. Damit entfällt die Notwendigkeit, Strubs für jede Übersetzung neu zu laden. Nach der Übersetzung wird mit »E« der Editbereich gewählt, dort das Objektprogramm geladen und getestet. Anschließend wird wieder das Quellprogramm in den Editbereich geladen, verbessert und mit »!« und »U« neu übersetzt und so weiter.

Bei Programmen, die nicht im Editbereich getestet werden können, handelt es sich erstens um Programme, die an einer festgelegten Stelle im Speicher stehen müssen. Strubs selbst ist solch ein Programm. Es muß unbedingt am normalen Basic-Anfang stehen. Solche Programme sind allerdings ziemlich selten. Häufiger dagegen findet sich der zweite Fall: Programme, die den Speicherbereich von 700 bis 800 verändern. Hier steht die in Folge 1 erwähnte Interpreter-Erweiterung von Strubs. Dadurch sind vor allem Programme betroffen, die in diesem Bereich Sprites oder Maschinenprogramme benutzen.

Kommen wir noch einmal auf das Schreiben und Editieren von Quellprogrammen zurück. Bisher haben wir nur davon gesprochen, daß die Programmtexte im Editbereich editiert wurden. Diese Methode hat insbesondere bei der Entwicklung umfangreicher Programme einen Nachteil: Da Strubs selbst mit einer Interpreter-Erweiterung arbeitet, kann man nicht gleichzeitig andere Interpreter-Erweiterungen — zum Beispiel Toolkits oder das DOS — benutzen. Möchte man auf Befehle wie MERGE, AUTO, FIND etc. nicht verzichten, dann kann man die Quellprogramme vollkommen unabhängig von Strubs entwickeln und erst anschließend das fertige Quellprogramm in den Editbereich laden.

Es zeigt sich, daß die meisten Beschränkungen bei der Arbeit mit Strubs ihren einzigen Grund in der kleinen Interpreter-Erweiterung haben. Wie bereits in der ersten Folge erwähnt, besteht der einzige Sinn dieser Erweiterung darin, das Starten von Strubs vom Editbereich aus dadurch bequemer zu gestalten, daß die Befehlsfolge

POKE 44,8: RUN

durch Eingabe von »!« abgekürzt werden kann. Wenn Sie bereit sind, diese Befehlsfolge jedesmal von Hand einzugeben, können Sie auf die Erweiterung verzichten, indem Sie im Programm die Zeilen 45600 bis 45680 einfach weglassen. Damit fallen dann die oben erwähnten Beschränkungen weg, das heißt die unter Fall 2 erwähnten Programme können im Editbereich getestet werden und Strubs kann zusammen mit einer Programmierhilfe benutzt werden. Aber editieren Sie keine Quellprogramme unter Simons Basic. Dazu sind weitere Anpassungen erforderlich, auf die wir später näher eingehen. Insbesondere wenn ein Programm aus fertigen Modulen zusammengesetzt werden soll, sind solche Programmierhilfen erforderlich.

Dieser Vorgang verläuft auf der Quellprogramm-Ebene aufgrund der Unabhängigkeit von Zeilennummern relativ einfach. Die einzelnen Programmteile werden in beliebiger Reihenfolge zusammengesetzt. Dazu kann ein Toolkit oder auch das kleine MERGE-Programm aus dem 64'er, Ausgabe 4/84 benutzt werden. Dabei können ruhig gleiche Zeilennummern auftreten und auch die Reihenfolge der Zeilennummern ist beliebig. Anschließend wird der fertige Programmtext mit Hilfe einer RENUMBER-Routine vernünftig durchnumeriert. Da Basic keine lokalen Variablen kennt, ist allerdings vor dem Zusammensetzen auf die Variablennamen zu achten. Um unerwünschte Seiteneffekte zu vermeiden, sind eventuell einige Umbenennungen vorzunehmen. Etwas aufwendiger gestaltet sich der Prozeß, wenn verschiedene Programmteile erst auf der Objekt-Ebene zusammengesetzt werden sollen. Hierbei ist darauf zu achten, daß sich die Bereiche der Zeilennummern nicht überschneiden. Weisen Sie jedem Programmteil einen bestimmten Zeilennummernbereich zu und verlegen diesen Teil gegebenenfalls mit RENUMBER in diesen Bereich. Anschließend werden nun in jedem Programmteil alle externen Routinen, die dieser Teil aufruft, mit Hilfe der EXTERN-Deklaration vereinbart (das sind die Routinen, die erst nach der Übersetzung angefügt werden). Das oben für die Variablennamen Gesagte gilt hier entsprechend. Jetzt können die einzelnen Teile getrennt übersetzt und anschließend in der richtigen Reihenfolge verknüpft werden.

Falls Sie keine Erweiterung besitzen, dann können Sie Strubs um eine RENUMBER-Routine erweitern: Fügen Sie die

Zeilen aus Bild 8 in das Objektprogramm von Strubs ein — und, falls Sie das Quellprogramm besitzen, dort entsprechend die Zeilen aus Bild 7. Um nun diese Routine anzubinden, müssen nur noch zwei Zeilen in das Menü eingefügt werden:
40110 PRINT "{CDOWN} {REV ON} R {REV OFF}ENUMBER"
und
40210 IF L$="R" THEN GOSUB 55000: GOTO 40050
beziehungsweise für das Quellprogramm:
40210 IF Z$="R" THEN GOSUB £RENUMBER: GOTO £MENUE

Diese Routine kann dann mit »R« aufgerufen werden, um ein Programm, das sich im Editbereich befindet, umzunumerieren. Das Programm Strubs arbeitet nicht mit der Datasette, sondern es benötigt eine Floppy. Damit der für Strubs unterhalb des Edit-Bereichs reservierte Platz nicht überschritten wird, ist darauf zu achten, daß beim Eintippen des Programms keine Blanks eingefügt werden.

Der Editbereich beginnt bei 40*256+1. Vor dem 1. Start von Strubs läßt sich mit PEEK(46) feststellen, ob Strubs diese Grenze nicht überschreitet (der Wert muß kleiner als 40 sein, im Originalprogramm liegt er bei 34).

Da Strubs den Zeiger für »Variablen-Anfang« heraufsetzt, sollte es immer von dem 1. Start abgespeichert werden (auch bei Veränderungen). Sollte man dies einmal vergessen, kann man durch »POKE 46,39:CLR« vor dem Abspeichern Strubs in die richtigen Grenzen verweisen. Die notwendigen Änderungen für VC 20 (mit 16 KByte) sind dem Bild 9 zu entnehmen. Hier beginnt der Editbereich bei 46*256+1.

Zusätzliche Funktionen

Die strukturierte Programmierung bietet vor allem Vorteile in bezug auf Wartung, Änderungen und Erweiterbarkeit von Programmen. Dies gilt auch für das Programm Strubs. Um in den Genuß dieser Vorteile zu gelangen, ist allerdings der Zugang zum Quellprogramm erforderlich. Wenn Sie sich das Objektprogramm ansehen, werden Sie feststellen, daß es auch nicht viel aussagekräftiger als ein unkommentiertes Assemblerlisting ist. Wenn Sie an der Entwicklung eigener Programmerweiterungen interessiert sind, sollten Sie sich deshalb beim Verlag das Quellprogramm besorgen. Da ich hier davon ausgehen muß, daß die meisten Leser das Quellprogramm nicht besitzen, lohnt es sich gar nicht erst, systematisch die einzelnen Programmteile vorzustellen.

Statt dessen wollen wir nur die für Programmerweiterungen wichtigsten Programmelemente vorstellen und anhand einiger exemplarischer Erweiterungen, die auch ohne sich weitere Gedanken zu machen, einfach eingetippt werden können, aufzeigen, wie man Erweiterungen implementieren kann und was dabei zu beachten ist. Aus dem gleichen Grund geben wir nur die Änderungen an, die im Objektprogramm vorzunehmen sind. Eine Anpassung an das Quellprogramm dürfte keine Probleme bereiten.

Achten Sie bei allen Programmänderungen darauf, daß das geänderte Programm abgespeichert wird, bevor es zum ersten Mal gestartet wird, da das Programm den Zeiger auf das Programmende verstellt. Sollte das Programm durch Erweiterungen so lang werden, daß es in den Editbereich hineinreicht, kann der Anfang des Editbereichs in Schritten zu 256 Byte nach oben verschoben werden, um Platz zu schaffen. Dazu ist in den Zeilen 70 bis 80 die Zahl 40 überall, wo sie auftaucht, durch eine größere Zahl (jeweils 4 für jedes Kilobyte) zu ersetzen.

Die wichtigsten Programmelemente

Eine grobe Übersicht über den Aufbau des Programms haben wir bereits gegeben. Bevor wir uns nun mit einzelnen Erweiterungen beschäftigen, wollen wir zunächst einmal die wichtigsten Programmelemente vorstellen, die man für Änderungen und Erweiterungen des Programms benötigt. Wie bereits erwähnt, liest Strubs das Quellprogramm zweimal vom Anfang bis zum Ende durch. Um Zeit zu sparen, wird im 1. Lauf nur jeweils der Anfang einer Zeile untersucht. Deshalb müssen alle Befehle, die bereits im 1. Lauf zu behandeln sind, auch am Anfang einer Zeile stehen, während Befehle, die nur im 2. Lauf behandelt werden, überall stehen können. Ein Beispiel:

Die Definition von Marken muß am Zeilenanfang erfolgen, während der Aufruf von Marken an jeder Stelle erfolgen kann. Die Aufgabe des 1. Laufs besteht darin, verschiedene Tabellen anzulegen, mit deren Hilfe dann im 2. Lauf das endgültige Objektprogramm erzeugt wird.

Jede dieser Tabellen besteht aus einem oder mehreren Array(s), einer Variablen, deren zweiter Buchstabe ein »M« für »Maximal« ist und die Dimension, das heißt die maximale Zahl von Einträgen festlegt, und aus einer Variablen, deren zweiter Buchstabe ein »P« für »Pointer« ist und die auf den jeweils nächsten freien Listenplatz zeigt. Bei Speicherplatzproblemen brauchen nur die Werte der Dimensionsvariablen im Init-Teil geändert zu werden. Möchte man zum Beispiel mehr als 99 Marken (die jetzige Maximalzahl) benutzen, dann schreibt man in Zeile 45060 zum Beispiel »MM=150:...«.

Die Tabellen werden in den Zeilen 45050 bis 45200 definiert (Bild 10). Die Dimension des Stacks bestimmt die mögliche Schachtelungstiefe. Dazu kommen die Tabellen der neuen Befehle (Zeile 45260 bis 45274) und der Fehlermeldungen (Zeile 45480 bis 45514).

Dem schrittweisen Lesen des Quellprogramms dienen die Variablen C und NC. Die Variable C enthält den Code des jeweils zuletzt gelesenen Zeichens, wobei der Wert 0 ein Zeilen-

```
40050 PRINT"J";"  *****************"
40052 PRINT "  * -- STRUBS  -- *"
40053 PRINT "  *  PRECOMPILER  *"
40055 PRINT "  * BITTE WAEHLEN *"
40058 PRINT "  *****************"
40060 PRINT"[illegible]EDIT"
```

Bild 9. Diese Änderungen sind für die Anpassung von Strubs an den VC 20 (mit mindestens 16 KByte Erweiterung) erforderlich.

```
45600 I=0:READW
45610 POKE704+I,W:I=I+1:READW:IFW<256THEN45610
45620 DATA32,115,0,8,201,33,240,4,40,76,231,199
45630 DATA169,18,133,44,169,138,76,231,199,999
```

```
15 '**  ---- STRUBS.4.QP ---      **

45050 ' ** MARKEN-TABELLE:
45060 £INIT: MM'AX'=99:DIM MA$(MM),MA%(MM):MP=0
45069 '
45120 '
45130 ' ** LOOP-TABELLE:
45131 ' *LO(..,0)=ZNR.LOOP
45132 ' *LO(..,1)=ZNR. ZUGEHOERIGES ELOOP
45135 L'OOP'M'AX'=140:DIM LO'OP'%(LM,1):L'OOP'P'OINTER'=0
45138 '
45140 ' ** IF-TABELLE:
45145 IM'AX'=270:DIM I%(IM):IP=0
45149 '
45188 '
45189 ' ** STACK:
45190 SM'AX'=60:DIM S'TACK'%(SM):SP'TR'=0
45200 '
READY.
```

Bild 10. Tabellen

ende markiert. Die Variable NC enthält die Adresse des nächsten zu lesenden Zeichens.

Im 2. Lauf wird zeilenweise das Objektprogramm erzeugt, wobei die jeweils aktuelle Zeile in der Variablen Z$ aufgebaut wird. Dabei enthalten die beiden ersten Zeichen von Z$ Low- und Highbyte der Zeilennummer (so wie sie später im Speicher steht), und das letzte Zeichen der fertigen Zeile besteht aus dem Zeichen CHR$(0).

Die relevanten Zeichencodes, auf die Strubs reagiert, werden in den Zeilen 45240 bis 45254 definiert (Bild 11). Die Variable ZA enthält die Adresse des Anfangs der Zeile, die gerade bearbeitet wird. In EA steht die Startadresse des Editbereichs.

Damit kommen wir zu den für Erweiterungen wichtigen Modulen von Strubs. Die Prozedur »NEXTCHAR« sucht ab Adresse NC das nächste relevante Zeichen des Quellprogrammtextes und liefert dessen Code in der Variablen C. Dabei werden Leerzeichen (Zeile 250) und Kommentare (Zeile 280-295) überlesen. Strings werden direkt in die Ausgabezeile Z$ übertragen (Zeile 350). Der Zeiger NC wird auf das nächste zu lesende Zeichen gesetzt. Die Prozedur »HOLNAME« (Zeile 750-830) liest ab aktueller Adresse NC einen Namen (zum Beispiel Befehl, Label), und zwar bis eines der Trennzeichen »:«, »,«, Blank oder Zeilenende erscheint. Der Name wird in der Variablen T$ ausgegeben, C enthält das erste relevante Zeichen hinter dem Namen (das ist außer beim Blank das Trennzeichen), und NC zeigt auf das nächste Zeichen.

Die Prozedur »SCHREIBZEILE« (Zeile 550-580) generiert auf der Diskette aus den nacheinander eingegebenen Zeilen Z$ das zusammenhängende Objektprogramm und gibt die Nummer der aktuellen Zeile auf dem Bildschirm aus. Die Variable AA (Linkadresse) darf außerhalb dieser Routine nicht verändert werden!

Die Prozedur »ERROR« (Zeile 8050 bis 8099) erwartet als Eingabe einen Fehlercode ER. Dabei handelt es sich um den Index der Fehlermeldung in der Tabelle der Fehlermeldungen. Die Zeilennummer und die Fehlermeldung werden auf dem Bildschirm ausgegeben und zugleich in eine Fehlertabelle eingetragen, die man sich nach der Übersetzung auf Bildschirm oder Drucker ausgeben lassen kann. Zusätzlich wird die Fehlermeldung in die Ausgabezeile Z$ geschrieben, so daß sie auch im Objektprogramm erscheint. Die Übersetzung wird mit der folgenden Zeile fortgesetzt.

Die Prozedur »ABBRUCH« (Zeile 50000 bis 50030) sorgt für einen kontrollierten Abbruch der Übersetzung. Sie erwartet ebenfalls als Eingabe den Fehlercode ER und gibt die entsprechende Fehlermeldung aus. Danach wird die Tabelle der bisher bemerkten Fehler ausgegeben, offene Files ordnungsgemäß geschlossen und Strubs neu gestartet.

Die Prozedur »WARTEN« (Zeile 49550 bis 49570) fordert den Benutzer auf, eine Taste zu drücken und wartet auf den Tastendruck.

Die Prozedur »INIT« (Zeile 45050 bis 45999) enthält die Definition der Variablen und Tabellen sowie die Interpretererweiterung.

Im »MENÜ« (Zeile 40050 bis 40495) können die verschiedenen Funktionen angewählt werden.

Die Prozeduren »BEFEHLE IM 1. LAUF« (Zeile 1550-2497) und »BEFEHLE IM 2. LAUF« (Zeile 2550-3640) werden von Strubs aufgerufen, sobald im Quellprogramm das Erkennungszeichen »!« für Befehle (Code in der Variablen BE) entdeckt wird. Sie holen den Namen des Befehls, suchen diesen in der Befehlstabelle und rufen entsprechend dem Index (+1) des Befehls in dieser Tabelle ein Unterprogramm auf. Falls der Befehl nicht in der Tabelle gefunden wird, wird eine entsprechende Fehlermeldung ausgegeben. Im 1. Lauf kommt noch die Ausgabe der Blockstruktur hinzu. Hierzu dient die Variable In (für Indentmodus). IN=0 bedeutet, auf der gleichen Schachtelungsebene zu bleiben.

Damit haben wir nun das notwendige Wissen zusammen, um an dem Programm Strubs einige Änderungen und Erweiterungen vorzunehmen.

Andere Anwendungen

Bei den Programmtexten, die Strubs übersetzt, handelt es sich zwar um erweiterte Basic-Programme, aber nichtsdestoweniger um Basic-Programme. Deshalb ist es relativ einfach, Strubs auch zur Bearbeitung ganz normaler Basic-Programme einzusetzen. Zwei sinnvolle Möglichkeiten wollen wir im folgenden vorstellen.

1. Ein SPEED-UP-Programm, um normale Basic-Programme schneller zu machen.
2. Ein Programm, das besser lesbare Listings erstellt.

Dabei ist zu beachten, daß die Änderungen, die wir dazu vornehmen, nicht wie die Makro-Funktion eine Erweiterung des eigentlichen Programmes Strubs und seiner Funktion darstellen, sondern daß wir zwei völlig neue Programme mit völlig neuen Aufgaben erhalten. Deshalb sollten auch die erhaltenen Programme unter neuen Namen, beispielsweise »SPEED-UP« und »LISTER«, abgespeichert werden. Das Arbeiten mit diesen Programmen unterscheidet sich nicht von der Arbeit mit dem »normalen« Strubs-Programm.

Schnellere Basic-Programme

Zunächst wollen wir Strubs so ändern, daß es normale Basic-Programme in Programme übersetzt, die keine Leerzeichen und Kommentare mehr enthalten und dadurch schneller ablaufen. Wie Sie sich erinnern werden, benutzt Strubs für Kommentare, die gelöscht werden sollen, ein eigenes Zeichen » ' «. Kommentare, die mit REM gekennzeichnet werden, bleiben im Objektprogramm erhalten. Da Strubs bereits alle Blanks entfernt (außer in Strings), brauchen wir nur noch dafür zu sorgen, daß Strubs auf das REM-Token reagiert wie bisher auf das Kommentarzeichen »'«. Die relevanten Zeichencodes, auf die Strubs reagiert, werden in den Zeilen 45250 bis 45254 definiert (Bild 11). Wir brauchen nur in Zeile 45250 das KO=ASC(»'«) durch KO=143 (143 ist das REM-Token) ersetzen, und schon ist das Speed-Up-Programm fertig. Genauso können Sie die Erkennungszeichen für Label und die neuen Befehle ändern. Dies ist, um Konflikte zu vermeiden, für den Fall sinnvoll, daß Sie mit Strubs Programme für Interpretererweiterungen übersetzen, die ihrerseits »!« oder das Pfundzeichen als Erkennungszeichen für ihre neuen Befehle benutzen.

```
45240 ' ** RELEVANTE ZEICHENCODES **
45250 DP=ASC(":"):KO'MMENTAR'=ASC("'"):LA'BEL'=ASC("£"):NU$=CHR$(0):BL=ASC(" ")
45253 BE'FEHL'=ASC("!"):TE'XT("")'=34:G'O'T'O-CODE'$=CHR$(137)
45254 I'F'C'ODE'$=CHR$(139):TH'EN-CODE'=167:NO'T'$=CHR$(168):K'OM'M'A-CODE'=44
```

Bild 11. Relevante Zeichencodes

Haben Sie in Ihren Listings häufig Grafik- und Steuerzeichen? Dann können Sie sich viel Ärger ersparen, wenn Sie das Listing vorher mit dem Programm »LISTER« aufbereiten. »LISTER« übersetzt Basic-Programme in Programmtexte, in denen die schwer entzifferbaren Steuer- und Grafikzeichen innerhalb von Strings durch lesbare Worte »⟨CDOWN«⟩ oder »⟨HOME⟩« ersetzt sind (Bild 12).

Listing

Dazu ändern wir eine Zeile innerhalb der Prozedur »NEXTCHAR«. In Zeile 350 werden gelesene Zeichen mit dem ASCII-Code C innerhalb von Strings direkt in die Ausgabezeile Z$ übertragen. Wenn wir nun in Zeile 350 Z$=Z$+CHR$(C) durch Z$=Z$+C$(C) ersetzen, dann können wir ein Array C$(255) definieren, das in jedem ASCII-Wert den String enthält, der dafür im Objektprogramm erscheinen soll. Die Definition dieses Arrays gehört in das Modul »INITIALISIERUNG«:

```
45300 DIM C$(255):FOR I=0 TO 255:C$(I)=CHR$(I):
NEXT
```

Damit haben wir zugleich unser Array mit den normalen Werten vorbesetzt. Jetzt bleiben nur noch

```
45310 C$(17)="⟨CDOWN⟩":C$(19)="⟨HOME⟩"
45312 C$(28)="⟨ROT⟩":C$(31)="⟨BLAU⟩"
... etc.
```

Hier können Sie nun jedem Zeichen ein beliebiges Wort zuordnen: Den ASCII-Code der einzelnen Zeichen finden Sie im C-64 Handbuch auf Seite 135 oder Sie können ihn einfach durch Eingabe von

```
PRINT ASC("X")
```

feststellen, wobei »X« für das interessierende Zeichen steht. Bei sehr vielen Zeichen innerhalb eines Strings kann es allerdings vorkommen, daß die Zeilen zu lang werden. Deshalb sollten die Worte möglichst kurz gewählt werden.

Makros

An einem etwas umfangreicheren Beispiel wollen wir nun zeigen, wie man neue Strubs-Befehle implementiert und wie man die Prozeduren von Strubs benutzen kann. Dies soll am Beispiel einer Makro-Funktion demonstriert werden.

Makros, vor allem von Assemblern her bekannt, stellen so etwas wie Abkürzungen für kurze Programmausschnitte dar. Dadurch verringert sich die Tipparbeit und vor allem werden die Quellprogramme übersichtlicher.

In der Makro-Definition wird ein Makro-Name definiert und diesem ein Programmstück zugeordnet. Überall, wo nun im Quellprogramm ein Makro aufgerufen wird, erscheint im Objektprogramm an dieser Stelle das entsprechende Programmstück. Ein einmal definiertes Makro kann wie ein Label beliebig oft aufgerufen werden.

Für die Definition eines Makros wollen wir den Befehl »!DMAKRO« und für den Aufruf eines Makros den Befehl »!M« wählen. Ein Beispiel mag die Wirkungsweise der neuen Befehle demonstrieren:

```
10 !DMAKRO:NAME SYS 833:X=PEEK (878)
...
200 PRINT X:!M,NAME:PRINT X
```

Die Definitionszeile 10 wird gelöscht, da sie nur für die Übersetzung notwendige Informationen enthält. Die Zeile 200 mit dem Makro-Aufruf sieht im Objektprogramm folgendermaßen aus:

```
200 PRINTX:SYS833:X=PEEK(878):PRINTX
```

Einige Beispiele für Makros und deren korrekte Benutzung sowie das sich ergebende Objektprogramm zeigt Bild 13. Vor allem ist darauf zu achten, daß Makronamen wie alle Befehls- und Labelnamen mit einem der Trennzeichen abgeschlossen werden müssen. Insbesondere darf bei der Makrodefinition und beim Aufruf mit nachfolgenden Parametern (Spritemakros in Zeile 120 und 130) nicht das Blank hinter dem Makronamen vergessen werden! Jede Makrodefinition benötigt eine eigene Zeile. Eine Übergabe von Parametern an ein Makro ist nicht möglich. Achten Sie bei der Arbeit mit Makros darauf, daß die entstehenden Zeilen des Objektprogramms nicht zu lang werden. Zeilen, die länger als 80 Zeichen sind, lassen sich nicht mehr editieren. Zeilen, die länger als 256 Zeichen werden, führen zum unkontrollierten Abbruch der Übersetzung mit »String too long error«. In diesem Fall kann man mit »GOTO 50000« die Nummer der verantwortlichen Zeile erfahren und offene Files schließen.

Um die Übersetzung zu ermöglichen, muß im 1. Lauf eine Tabelle der Makronamen und der zugehörigen Programmausschnitte angelegt werden. Im 2. Lauf werden dann alle Aufrufe durch den zugehörigen Text ersetzt. Die Verteilung auf zwei Läufe bietet den Vorteil, daß ein Makro (ebenso wie Label) auch schon vor der Definition aufgerufen werden kann.

Zur Implementation sind folgende Schritte erforderlich: Zunächst muß dem Übersetzungsprogramm mitgeteilt werden, daß es zwei neue Befehle gibt. Dann müssen wir die notwendige Tabelle definieren und auch entsprechende Fehlermeldungen vorsehen. Diese Erweiterungen gehören in den INIT-Teil.

Schließlich muß noch dafür gesorgt werden, daß Strubs weiß, wie es im 1. und 2. Lauf auf die neuen Befehle zu reagieren hat. Die Befehlstabelle wird in Zeile 45265 definiert. Hier erhöhen wir die Zahl der Befehle um 2 und fügen dann noch eine DATA-Zeile mit den beiden neuen Befehlsnamen ein:

```
45265 BM=15:...
45275 DATA DMAKRO,M
```

Wählt man Befehlsnamen, die reservierte Basic-Worte enthalten, dann müssen die Tokens berücksichtigt werden (wie dies für IF in der Zeile 45271 geschieht). Für einen Befehl »DEFMAKRO« wäre zum Beispiel BE$(14)=CHR$(150)+"MAKRO" zu setzen (150=DEF-Token).

Für die Tabelle wählen wir ein Array NA$(NM,1), da der Name M bereits für die Markentabelle vergeben ist. Die Dimension (..,0) soll die Namen und die Dimension (..,1) den zugehörigen Text aufnehmen.

```
45155 NM=40:DIM NA$(NM,1):NP=0
```

Damit können 41 Makros definiert werden. Indem wir die Zahl der Fehlermeldungen von 9 auf 11 erhöhen, erhalten wir die beiden neuen Fehlercodes 10 und 11 für »zu viele Makros« und »undefiniertes Makro«.

```
45480 EM=11:DIM...
45500 FOR I=0 TO EM:READ ...
45515 DATA "ZU VIELE MAKROS", "UNDEFINIERTES
MAKRO"
```

```
10 REM ****  LISTER-DEMO **************
30 PRINT"[illegible]TEST[illegible]"
20 REM WIRD ZU:
30 PRINT"<CD><CR><CU><CL><RON><ROF><HO>TEST<DEL><INS><WHT><RED><GRN><BLU><BLK><P
UR><YEL><CYN>"
READY.
```

Bild 12. Beispiellister

```
10 REM ***** MAKROS BEISPIELE ********
20 !DMAKRO:STOPAN POKE 788,49
30 !DMAKRO:STOPAUS POKE 788,52
40 !DMAKRO:READJOY JS=PEEK(56320)
50 !DMAKRO:WARTE POKE 198,0:WAIT 198,1
60 !DMAKRO:CLOSEALL SYS 65511
70 !DMAKRO:SPRITEXPOS POKE 53248+2*
80 !DMAKRO:SPRITEYPOS POKE 53249+2*
90 !DMAKRO:SPRITEAN POKE 53269,PEEK(53269) OR 2↑
95 '
96 '  ********  AUFRUFE:  ************
100 !M,STOPAUS:!M,CLOSEALL:!M,STOPAN
110 !M,READJOY:PRINT JS
120 !M,SPRITEAN 5:!M,WARTE
130 !M,SPRITEXPOS 5,240:!M,SPRITEYPOS 5,170
140 !M,FEHLER

READY.

10 REM*****MAKROSBEISPIELE********
100 POKE788,52:SYS65511:POKE788,49
110 JS=PEEK(56320):PRINTJS
120 POKE53269,PEEK(53269)OR2↑5:POKE198,0:WAIT198,1
130 POKE53248+2*5,240:POKE53249+2*5,170
140 ***** ERR:UNDEFINIERTES MAKRO********

READY.
```

Bild 13. Beispiele Makros

```
0 ********* SPEED-UP ****************
45250 DP=ASC(":"):KO=143:LA=ASC("£"):NU$=CHR$(0):BL=ASC(" ")

READY.

0 ****** LISTER   ********************
350 Z$=Z$+C$(C):NC=NC+1:C=PEEK(NC):IF C AND C<>TE THEN 350
45300 DIM C$(255):FOR I=0 TO 255:C$(I)=CHR$(I):NEXT
45310 C$(17)="<CD>":C$(29)="<CR>":C$(145)="<CU>":C$(157)="<CL>"
45312 C$(18)="<RON>":C$(146)="<ROF>":C$(19)="<HO>":C$(147)="<CLR>"
45314 C$(20)="<DEL>":C$(148)="<INS>"
45316 C$(5)="<WHT>":C$(28)="<RED>":C$(30)="<GRN>":C$(31)="<BLU>"
45318 C$(144)="<BLK>":C$(156)="<PUR>":C$(158)="<YEL>":C$(159)="<CYN>"

READY.

0 ********** MAKROS *****************
1571 IF I>14 THEN ON I-14 GOSUB 2350,2360
2350 IF NP>NM THEN ER=10:GOTO 50000
2355 Z$="":GOSUB 750:NA$(NP,0)=T$
2360 Z$=Z$+CHR$(C):GOSUB 250:IF C<>0 THEN 2360
2370 NA$(NP,1)=Z$:NP=NP+1:IN=0
2375 RETURN
2380 IN=0:RETURN
2571 IF I>14 THEN ON I-14 GOSUB 3700,3750
3700 Z$="":C=0:RETURN
3750 GOSUB 750
3755 FOR I=0 TO NP:IF NA$(I,0)<>T$ THEN NEXT
3760 IF I>=NP THEN ER=11: GOTO 8050
3765 Z$=Z$+NA$(I,1):RETURN
45155 NM=40:DIM NA$(NM,1):NP=0
45265 BM=15:DIM BE$(BM)
45275 DATA DMAKRO,M
45480 EM=11:DIM ER%(30,1):EP=0:DIM ER$(EM)
45500 FOR I=0 TO EM:READ ER$(I):NEXT
45515 DATA "ZU VIELE MAKROS","UNDEFINIERTES MAKRO"
```

Bild 14. Die besprochenen Erweiterungen auf einen Blick

Nun müssen wir in die beiden Module »BEFEHLE IM 1. LAUF« beziehungsweise »BEFEHLE IM 2. LAUF« jeweils zwei Routinen für die neuen Befehle einfügen. Da die beiden Verteilerzeilen bereits voll sind, legen wir zwei neue Verteilerzeilen an, die dann aber auch gleich für zehn weiter, neue Befehle Platz bieten:
1571 IF I>14 THEN ON I-14 GOSUB 2350,2380
für den 1. Lauf und
2571 IF I>14 THEN ON I-14 GOSUB 3700,3750
für den 2. Lauf.

Die Routine für »!DMAKRO« im 1. Lauf soll prüfen, ob noch Platz in unserer Makro-Tabelle ist und, falls nicht, mit entsprechendem Fehlercode die Abbruch-Routine anspringen:
2350 IF NP>NM THEN ER=10:GOTO 50000

```
----------------------------------------------------------------------
CROSS REFERENCE MAP                    STRUBS.4.OP                PAGE
----------------------------------------------------------------------
AA      560*  565   570   5120* 5130
AD()    555   1125  1574  1605  2100  2260  5052  5920  6100  6550
6655  8050  8060  45410 50008
B$      1565* 1575  1577  1579  1581  2685* 2693  48060* 48070 49060*
49070 49560*
BE      3470  45253*
BE$()   1550  45641 45265 45270* 45271*
BL      250   295   795   820   45250*
BM      1565  45641 2565  45265* 45270
C       260*  265   280*  290*  295   350*  795*  800   820   2420
2425  2470  2590  3010  3036* 3400* 3470  3495* 3610  4100
4108  4115  4130  4360  4380  5580* 5585  8090*
C$      8090*
DI      1170  1605  2100  2260  2640  2693  3030  3090  3490  3630
6100  8060  45220* 48140 49140
DP      795   5585  45250*
E       5090* 5095
E$      5090* 5095
EA      80*   5052  5120  5555  6550  8950
EM      8060  45480*
EP      5180  8060* 45480* 49050 49110 49120
ER      1160* 1565* 1600* 1605* 1640* 2010* 2040* 2100* 2160* 2275*
2410* 2425* 2565* 5143* 6050* 8050  8060  8080  50008
ER$()   8050  8080  45480 45500* 49140 50008
ER%()   8060* 45480 49140
F$      5070* 5080
OT$     2640  2685  3090  45253*
H       1125* 1170* 1180  2260* 2300  2425* 2647* 2648  3100* 3130
3190* 3200  48055* 48102* 48150 49055* 49102* 49150
H%      565*  570
I       1140* 1160  1170  1550  1565  1569* 1570  2275* 2290  2300
45641 2565  2570  45270* 45500* 45600* 45610* 45650* 48120* 48140
48150 49120* 49140 49150
I%()    2100* 2300* 3030  3090  45145
IC$     3010  3600  45254* 45271
IM      45145*
IN      1575  1577  1579  1581  1615* 1660* 1680* 2025* 2052* 2107*
2240* 2320* 2485* 3620*
IP      2010* 2050* 3030  3036* 3090  3140* 6560* 45145*
KM      795   45254*
KO      265   280   6585  45250*
L       2647* 2648  3100* 3130  3190* 3200
LA      4100  4360  45250*
LM      1605  45135*
LO%()   1605* 2640  2693  3490  3630  45135
LP      1605* 2595* 6560* 45135*
MA$()   1140  6100* 45060 48140
MA%()   1170  6100* 45060 48140
MM      2410  6050  45060*
MP      1140  1160  2410  2460* 6050  6100* 45060* 48050 48120
NC      250*  260   280*  290*  350*  370*  795   800*  820*  4080*
4115* 5580*
NO$     3010  3600  45254*
NU$     2640  3090  4115  8090  45250*
S%()    2050* 2100  2275  2595* 2640  2693  3490  3630  45190
SM      1600  2010  2160  45190*
SP      1600  1605* 1640* 2010* 2040  2050  2100  2160* 2275  2595*
2630* 2640  2693  3490  3620* 3630  5143  6560* 45190*
T$      750*  800*  1120  1140  1550  45641 6100
TA      1575  1577* 1579  1581* 5136*
TE      350   45253*
TH      3030  3480  3630  45254*
W       45600* 45610* 45650*
X       45410
Z$      350*  550   560   570   1180* 2590* 2640* 2648* 2685  2693*
3010* 3030* 3090* 3130* 3200* 3400* 3470* 3480* 3490* 3600*
3610* 3630* 4060* 4108* 4115* 4380* 5096* 5099  8080* 40160*
40170 40180 40195
Z1      6550* 6655* 6660
ZA      555   1125  1574  1605  2100  2260  2647  3100  3190  4060
4080  5555* 5570  5580  5920* 6100  6550* 6585  6655* 8050
8060  50008
```

Bild 15. Variablenliste

Jetzt können wir mit Hilfe der Prozedur »HOLNAME« den Makro-Namen lesen und in unserer Tabelle speichern:
2355 Z$="":GOSUB750:NA$(NP,0)=T$

Nun übertragen wir den Rest der Definitionszeile mit Hilfe von »NEXTCHAR« nach Z$ (dadurch werden auch Strings mit übertragen. Als Ausgabezeile dient Z$ ja erst im 2. Lauf).
2360 Z$=Z$+CHR$(C):GOSUB 250:IF C<>0 THEN 2360

Nun brauchen wir nur noch den Text in die Tabelle aufzunehmen, den Zeiger zu erhöhen und den Indentmodus angeben.
2370 NA$(NP,1)=Z$:NP=NP+1:IN=0
2375 RETURN

Der Aufruf eines Makros interessiert im 1. Lauf nicht, also:
2380 IN=0:RETURN

Im 2. Lauf soll die Definitionszeile gelöscht werden. Dazu löschen wir den Ausgabestring und weisen C den Code für Zeilenende zu:
3700 Z$="":C=0:RETURN

Beim Aufruf eines Makros mit »!M« holen wir zunächst den Namen des Makros mit »HOLNAME« und suchen ihn in der Tabelle:

3750 GOSUB 750
3755 FOR I=0 TO NP: IF NA$(I,0)<>T$ THEN NEXT

Falls der Name nicht gefunden wird, erfolgt ein Sprung zur Error-Routine mit dem Code für »undefiniertes Makro«:

3760 IF I>NP THEN ER=11: GOTO 8050

Nun ist nur noch das definierte Programmstück in die Ausgabezeile zu übertragen:

3760 Z$=Z$+NA$(I,1):RETURN

Dadurch, daß diese Makro-Erweiterung Zeile für Zeile besprochen wurde, um zu zeigen, wie man die von Strubs vorgegebenen Prozeduren benutzen kann, ist vielleicht der Eindruck entstanden, eine solche Erweiterung sei relativ kompliziert. Wenn Sie sich aber das Ganze noch einmal genauer ansehen, können Sie feststellen, daß für die Implementation neuer Befehle im Prinzip nur drei Schritte erforderlich sind:

1. Eintrag der neuen Befehlsnamen in die Befehlstabelle
2. Einfügen der entsprechenden Routinen
3. Eintrag der Adressen dieser Routinen in die beiden Verteilerzeilen

Die ganze Arbeit des Suchens und Decodierens übernimmt Strubs automatisch.

Wie neue Funktionen (beispielsweise die Ausgabe der Makro-Tabelle) in das Menü aufgenommen werden können, haben Sie bereits in der letzten Folge am Beispiel der RENUMBER-Funktion gesehen.

Eine Zusammenstellung der oben besprochenen Erweiterungen finden Sie in Bild 14.

Strubs und Interpretererweiterungen

Wollen Sie mit Strubs Programme für Interpretererweiterungen bearbeiten, dann sind einige weitere Dinge zu beachten. Entfernen Sie zunächst die Interpretererweiterung von Strubs.

Falls die Erweiterung, die Sie benutzen wollen, nicht in den Editor eingreift, sondern ihre neuen Befehle durch besondere Zeichen (meistens »!«) gekennzeichnet werden, dann ändern Sie wie bereits oben beschrieben die entsprechenden Erkennungszeichen, die Strubs benutzt.

Bei Erweiterungen wie Simons Basic, die in den Editor eingreifen und die neuen Befehle wie der Basic-Interpreter durch eigene Tokens darstellen, ist es am einfachsten, den Strubs-Befehlen, deren Namen solche Befehle enthalten, neue Namen zu geben. Im Fall von Simons Basic sind davon beispielsweise Strubs-Befehle wie »!REPEAT«, »!UNTIL« oder »!ELSE« etc. betroffen.

Dazu sind nur die Namen in den DATA-Zeilen 45272 bis 45274 zu ändern. Sie können die betroffenen Strubs-Befehle aber auch wie oben am Beispiel von »DEFMAKRO« beschrieben aus den Tokens zusammensetzen. Dabei ist aber zu berücksichtigen, daß die Token von Simons Basic aus zwei Zeichen und nicht wie die normalen Token aus nur einem Zeichen bestehen.

Eine Liste der von Strubs benutzten Variablen bietet Bild 15. Dabei kennzeichnet das Zeichen »*« Zeilennummern, in denen eine Wertzuweisung an die Variable erfolgt.

Um sich an die Arbeitsweise von Strubs zu gewöhnen, können Sie das Quellprogramm »Menü« (Listing 2) eingeben. »Menü« faßt immer zehn Programme auf einer Bildschirmseite zusammen, die man mit den Cursor-Tasten durchblättern kann. Bei Programmen, die größer sind als »Menü«, muß in der ersten Zeile der Pointer auf das Basic-Ende korrigiert werden (siehe Zeile 1 von »Menü«, Listing 3). Hängt man bei Programmen, die von »Menü« geladen werden sollen, an das Programmende LOAD"Menü.obj" ,8 dann wird bei Programmende »Menü« automatisch geladen und gestartet.

(Matthias Törk/og)

```
5 REM STRUBS4/4.9.83                              <221>
51 PRINT"{CLR}";TAB(10);"********************
   "                                              <189>
52 PRINT TAB(10);"* --STRUBS.4 -- *"              <154>
55 PRINT TAB(10);"*{3SPACE}M.TOERK{5SPACE}
   *"                                             <165>
57 PRINT TAB(10);"* 4352 HERTEN{3SPACE}*"         <079>
58 PRINT TAB(10);"******************"             <181>
70 IF NOT(PEEK(46)<40 OR(PEEK(46)=40 AND P
   EEK(45)<3))THEN 75                             <103>
73 POKE 46,40:POKE 45,3:POKE 40*256,0:CLR         <062>
75 :                                              <051>
80 EA=40*256+1                                    <094>
100 GOSUB 45060                                   <057>
140 GOTO 40050                                    <193>
250 IF PEEK(NC)=BL THEN NC=NC+1:GOTO 250          <032>
260 C=PEEK(NC)                                    <054>
265 IF C<>KO THEN 320                             <147>
280 NC=NC+1:C=PEEK(NC):IF C AND C<>KO THEN
     280                                          <222>
290 IF C THEN NC=NC+1:C=PEEK(NC)                  <017>
295 IF C=BL THEN 250                              <038>
320 IF C<>TE THEN NC=NC+1:RETURN                  <124>
350 Z$=Z$+CHR$(C):NC=NC+1:C=PEEK(NC):IF C
    AND C<>TE THEN 350                            <164>
370 NC=NC+1                                       <017>
390 RETURN                                        <194>
550 IF LEN(Z$)<4 THEN RETURN                      <013>
555 PRINT FN AD(ZA+2)                             <060>
560 AA=AA+LEN(Z$)+2                               <075>
565 H%=AA/256                                     <112>
570 PRINT#1,CHR$(AA-256*H%);CHR$(H%);Z$;          <206>
580 RETURN                                        <130>
750 T$=""                                         <091>
790 :                                             <004>
795 C=PEEK(NC):IF C=DP OR C=KM OR C=BL OR
    C=0 THEN 811                                  <200>
800 NC=NC+1:T$=T$+CHR$(C)                         <126>
810 GOTO 790                                      <168>
811 :                                             <025>
820 NC=NC+1:IF C=BL THEN GOSUB 250                <050>
830 RETURN                                        <126>
1050 GOSUB 750                                    <106>
1120 IF NOT(T$="THIS")THEN 1131                   <252>
1125 H=FN AD(ZA+2)                                <115>
1130 GOTO 1175                                    <103>
1131 :                                            <091>
1140 FOR I=0 TO MP:IF MA$(I)<>T$THEN NEXT         <060>
1160 IF I>MP THEN ER=2:GOTO 8050:                 <110>
1170 H=MA%(I)+DI                                  <164>
1175 :                                            <135>
1180 Z$=Z$+MID$(STR$(H),2)                        <102>
1190 RETURN                                       <232>
1550 GOSUB 750                                    <098>
1560 FOR I=0 TO BM:IF T$<>BE$(I)THEN NEXT         <098>
1565 IF I>BM THEN ER=0:GOTO 8050                  <237>
1567 B$=BE$(I):IF I=3 THEN B$="IF"                <189>
1569 I=I+1                                        <028>
1570 ON I GOSUB 1600,1680,1640,2010,2040,2
     100,2160,2210,2260,2400,1710,1740,181
     0,1860                                       <187>
1574 PRINT FN AD(ZA+2);                           <242>
1575 IF IN=0 THEN PRINT TAB(TA);B$:RETURN         <241>
1577 IF IN=1 THEN PRINT TAB(TA);B$:TA=TA+1
     :RETURN                                      <094>
1579 IF IN=2 THEN PRINT TAB(TA-1);B$:RETUR
     N                                            <048>
1581 IF IN=3 THEN TA=TA-1:PRINT TAB(TA);B$
     :RETURN                                      <235>
1586 RETURN                                       <120>
1600 IF SP>SM THEN ER=3:GOTO 50000                <031>
1605 IF LP>LM THEN ER=5:GOTO 50000                <075>
1610 S%(SP)=LP:SP=SP+1:LO%(LP,0)=FN AD(ZA+
     2)-DI:LP=LP+1                                <122>
1615 IN=1:RETURN                                  <024>
1640 SP=SP-1:IF SP<0 THEN ER=1:GOTO 50000         <077>
1650 LO%(S%(SP),1)=FN AD(ZA+2)-DI                 <173>
1660 IN=3:RETURN                                  <133>
1680 IN=0:RETURN                                  <057>
1710 GOSUB 1600:RETURN                            <153>
1740 GOSUB 1640:RETURN                            <055>
1810 GOSUB 1600:RETURN                            <255>
1860 GOSUB 1640:RETURN                            <177>
2010 IF SP>SM THEN ER=3:GOTO 50000                <187>
2011 IF IP>IM THEN ER=4:GOTO 50000                <250>
```

Listing 1. Das Objektprogramm Strubs.
Bitte beachten Sie die Eingabehinweise auf Seite 16.

```
2020 S%(SP)=IP:IP=IP+1:SP=SP+1                    <119>
2025 IN=1:RETURN                                  <180>
2040 IF SP<1 THEN ER=1:GOTO 50000                 <045>
2041 IF IP>IM THEN ER=4:GOTO 50000                <024>
2044 I%(S%(SP-1))=FN AD(ZA+2)+1-DI                <052>
2050 S%(SP-1)=IP:IP=IP+1                          <167>
2052 IN=2:RETURN                                  <241>
2100 IF SP<1 THEN ER=1:GOTO 50000                 <107>
2105 SP=SP-1:I%(S%(SP))=FN AD(ZA+2)-DI            <214>
2107 IN=3:RETURN                                  <072>
2160 IF SP>SM THEN ER=3:GOTO 50000                <083>
2165 S%(SP)=-1:SP=SP+1                            <042>
2170 GOSUB 2010                                   <134>
2180 IN=1:RETURN                                  <081>
2210 GOSUB 2040                                   <014>
2230 GOSUB 2010                                   <194>
2240 IN=2:RETURN                                  <173>
2260 H=FN AD(ZA+2)-DI                             <020>
2270 :                                            <214>
2275 IF SP<1 THEN ER=1:GOTO 50000                 <026>
2280 SP=SP-1:I=S%(SP)                             <210>
2290 IF I<0 THEN 2311                             <022>
2300 I%(I)=H                                      <240>
2310 GOTO 2270                                    <228>
2311 :                                            <001>
2320 IN=3:RETURN                                  <031>
2400 :                                            <090>
2410 IF MP>MM THEN ER=6:GOTO 50000                <206>
2415 IF C AND C<>LA THEN GOSUB 250:GOTO 24
     15                                           <082>
2420 IF C THEN GOSUB 750                          <144>
2423 IF C THEN GOSUB 250                          <082>
2425 IF C<48 OR C>57 THEN ER=9:GOTO 8050          <133>
2430 MA$(MP)=T$:H=C                               <138>
2440 GOSUB 750                                    <226>
2450 MA%(MP)=VAL(CHR$(H)+T$)-DI                   <129>
2460 MP=MP+1                                      <211>
2470 IF C=0 THEN 2481                             <155>
2480 GOTO 2400                                    <206>
2481 :                                            <171>
2485 IN=0:RETURN                                  <100>
2550 GOSUB 750                                    <080>
2560 FOR I=0 TO BM:IF T$<>BE$(I)THEN NEXT         <082>
2565 IF I>BM THEN ER=0:GOTO 8050                  <221>
2568 I=I+1                                        <011>
2570 ON I GOSUB 2590,2685,2630,3010,3090,3
     190,3260,3310,3360,3400,3450,3550,358
     0,3600                                       <003>
2575 RETURN                                       <093>
2590 IF C=0 THEN Z$=Z$+":"                        <097>
2595 S%(SP)=LP:SP=SP+1:LP=LP+1                    <116>
2597 RETURN                                       <115>
2630 SP=SP-1                                      <216>
2640 Z$=Z$+GT$+MID$(STR$(LO%(S%(SP),0)+DI)
     ,2)+NU$                                      <178>
2642 GOSUB 550                                    <158>
2647 L=PEEK(ZA+2)+1:H=PEEK(ZA+3):IF L>255
     THEN L=0:H=H+1                               <148>
2648 Z$=CHR$(L)+CHR$(H)+":"                       <009>
2650 RETURN                                       <168>
2685 B$="":IF RIGHT$(Z$,1)<>CHR$(167)THEN
     B$=GT$                                       <013>
2693 Z$=Z$+B$+MID$(STR$(LO%(S%(SP-1),1)+DI
     +1),2)                                       <004>
2695 RETURN                                       <213>
3010 Z$=Z$+IC$+NO$+"("+CHR$(C)                    <073>
3020 GOSUB 250:IF C<>TH AND C THEN Z$=Z$+C
     HR$(C):GOTO 3020                             <003>
3030 Z$=Z$+")"+CHR$(TH)+MID$(STR$(I%(IP)+D
     I),2)                                        <004>
3036 IP=IP+1:C=0:RETURN                           <138>
3090 Z$=Z$+GT$+MID$(STR$(I%(IP)+DI),2)+NU$        <233>
3100 GOSUB 550                                    <108>
3120 L=PEEK(ZA+2)+1:H=PEEK(ZA+3):IF L>255
     THEN L=0:H=H+1                               <113>
3130 Z$=CHR$(L)+CHR$(H)+":"                       <239>
3140 IP=IP+1:RETURN                               <159>
3190 L=PEEK(ZA+2):H=PEEK(ZA+3)                    <141>
3200 Z$=CHR$(L)+CHR$(H)+":"                       <053>
3210 RETURN                                       <220>
3260 GOSUB 3010:RETURN                            <131>
3310 GOSUB 3090                                   <011>
3320 Z$=LEFT$(Z$,LEN(Z$)-1)                       <172>
3330 GOSUB 3010                                   <032>
3340 RETURN                                       <096>
3360 GOSUB 3190                                   <022>
3370 RETURN                                       <126>
```

```
3400 Z$="":C=0:RETURN                             <033>
3450 GOSUB 2590                                   <225>
3460 Z$=Z$+IC$+NO$+"("                            <213>
3470 IF C<>BE AND C THEN Z$=Z$+CHR$(C):GOS
     UB 250:GOTO 3470                             <039>
3480 Z$=Z$+")"+CHR$(TH)                           <068>
3490 Z$=Z$+MID$(STR$(LO%(S%(SP-1),1)+DI+1)
     ,2)                                          <139>
3495 C=0:RETURN                                   <080>
3550 GOSUB 2630:RETURN                            <063>
3580 GOSUB 2590:RETURN                            <014>
3600 Z$=Z$+IC$+NO$+"("                            <099>
3610 IF C THEN Z$=Z$+CHR$(C):GOSUB 250:GOT
     O 3610                                       <018>
3620 SP=SP-1:IN=3                                 <090>
3630 Z$=Z$+")"+CHR$(TH)+MID$(STR$(LO%(S%(S
     P),0)+DI),2)                                 <253>
3640 RETURN                                       <142>
4060 Z$=CHR$(PEEK(ZA+2))+CHR$(PEEK(ZA+3))         <029>
4080 NC=ZA+4:GOSUB 250                            <068>
4090 IF C=DP THEN GOSUB 250                       <064>
4100 IF NOT(C=LA)THEN 4110                        <106>
4105 GOSUB 750:IF C=DP THEN GOSUB 250             <128>
4108 IF C=0 THEN Z$=Z$+":"                        <091>
4110 :                                            <022>
4115 NC=NC-1:IF C=0 THEN Z$=Z$+NU$                <064>
4130 :IF C=0 THEN 4397                            <038>
4132 GOSUB 250                                    <100>
4150 IF NOT(C=BE)THEN 4359                        <125>
4155 GOSUB 2550                                   <039>
4358 GOTO 4378                                    <022>
4359 :                                            <017>
4360 IF C=LA THEN GOSUB 1050                      <164>
4378 :                                            <036>
4380 Z$=Z$+CHR$(C)                                <009>
4396 GOTO 4130                                    <154>
4397 :                                            <055>
4398 RETURN                                       <138>
5050 PRINT"{CLR,3SPACE}***** UEBERSETZEN{4
     SPACE}****{3DOWN}"                           <252>
5052 IF NOT(FN AD(EA)<EA+5 OR FN AD(EA)>EA
     +83)THEN 5054                                <110>
5053 PRINT"KEIN PROGRAMM VORHANDEN":GOSUB
     49550:RETURN                                 <004>
5054 :                                            <204>
5058 PRINT"BITTE DISK EINLEGEN{SPACE,2DOWN
     ,SPACE}"                                     <196>
5060 PRINT"NAME FUER OBJEKT-PROGRAMM"             <187>
5065 POKE 198,1:POKE 631,34                       <107>
5070 INPUT F$                                     <126>
5080 OPEN 1,8,1,F$+",P,W":OPEN 15,8,15            <160>
5090 INPUT#15,E,E$:IF E=0 THEN 5101               <028>
5095 PRINT"DISK ERR:";E;E$                        <113>
5096 INPUT"NEUER VERSUCH";Z$                      <082>
5098 CLOSE 1:CLOSE 15                             <184>
5099 IF Z$<>"J"THEN RETURN                        <230>
5100 GOTO 5060                                    <182>
5101 :                                            <251>
5120 AA=EA                                        <090>
5130 PRINT#1,CHR$(AA AND 256);CHR$(AA/256)
     ;                                            <012>
5135 PRINT"1.LAUF"                                <162>
5136 TA=7                                         <164>
5140 GOSUB 5555                                   <097>
5143 IF SP>0 THEN PRINT SP;:ER=8:GOTO 5000
     0                                            <090>
5145 PRINT"2.LAUF"                                <188>
5150 GOSUB 6550                                   <050>
5160 PRINT#1,CHR$(0);CHR$(0);                     <012>
5180 CLOSE 1:PRINT"{DOWN}**";EP;" ERRORS *
     *":GOSUB 49550                               <086>
5190 RETURN                                       <168>
5555 ZA=EA                                        <115>
5570 IF NOT(ZA<>0)THEN 5931                       <037>
5580 NC=ZA+4:C=PEEK(NC):NC=NC+1                   <077>
5585 IF C=DP THEN GOSUB 250                       <035>
5590 IF C=LA THEN GOSUB 6050:IF C=DP THEN
     GOSUB 250                                    <137>
5620 IF C=BE THEN GOSUB 1550                      <108>
5920 ZA=FN AD(ZA)                                 <140>
5930 GOTO 5570                                    <108>
5931 :                                            <065>
5935 RETURN                                       <151>
6050 IF MP>MM THEN ER=6:GOTO 50000                <034>
6070 GOSUB 750                                    <044>
6100 MA$(MP)=T$:MA%(MP)=FN AD(ZA+2)-DI:MP=
     MP+1                                         <058>
```

```
6120 RETURN                                    <080>
6550 ZA=EA:Z1=FN AD(ZA)                        <148>
6560 LP=0:SP=0:IP=0                            <149>
6580 :                                         <206>
6585 IF NOT(PEEK(ZA+4)<>KO)THEN 6650           <098>
6590 GOSUB 4060                                <156>
6600 GOSUB 550                                 <050>
6650 :                                         <020>
6655 ZA=Z1:Z1=FN AD(Z1)                        <140>
6660 IF NOT(Z1=0)THEN 6580                     <084>
6680 RETURN                                    <134>
8050 PRINT"ERROR IN";FN AD(ZA+2),ER$(ER)       <224>
8060 IF EP<EM THEN ER%(EP,0)=FN AD(ZA+2)-D
     I:ER%(EP,1)=ER:EP=EP+1                    <227>
8080 Z$=LEFT$(Z$,2)+"***** ERR:"+ER$(ER)+"
     ********"                                 <174>
8090 C$=NU$:C=0                                <108>
8099 RETURN                                    <027>
8860 PRINT"{CLR,5DOWN}"                        <057>
8870 PRINT TAB(9);"**********************"     <091>
8880 PRINT TAB(9);"** ZURUECK MIT:{4SPACE}
     **"                                       <143>
8882 PRINT TAB(9);"**  ' ! '  [RETURN]{2SPAC
     E}**"                                     <180>
8940 PRINT TAB(9);"**********************"     <161>
8950 POKE 44,EA/256:POKE EA-1,0:CLR:END        <080>
8990 END                                       <102>
40050 PRINT"{CLR}";TAB(10);"**************
      ***"                                     <053>
40052 PRINT TAB(10);"* -- STRUBS{2SPACE}--
      *"                                       <171>
40053 PRINT TAB(10);"*{2SPACE}PRECOMPILER{
      2SPACE}*"                                <222>
40055 PRINT TAB(10);"* BITTE WAEHLEN *"        <205>
40058 PRINT TAB(10);"*****************"        <046>
40060 PRINT"{3DOWN,RVSON}E{RVOFF}DIT"          <250>
40070 PRINT"{DOWN,RVSON}U{RVOFF}EBERSETZEN
      "                                        <031>
40080 PRINT"{DOWN,RVSON}M{RVOFF}ARKEN-TABE
      LLE AUSGEBEN"                            <174>
40090 PRINT"{DOWN,RVSON}F{RVOFF}EHLER-TABE
      LLE AUSGEBEN"                            <043>
40100 PRINT"{DOWN,RVSON}S{RVOFF}CHLUSS"        <153>
40160 GET Z$:IF Z$=""THEN 40160                <101>
40170 IF Z$="E"THEN 8860                       <096>
40180 IF Z$="U"THEN GOSUB 5050:GOTO 40050      <164>
40190 IF Z$="S"THEN SYS 64738                  <173>
40195 IF Z$="M"THEN GOSUB 48050:GOTO 40050     <141>
40200 IF Z$="F"THEN GOSUB 49050:GOTO 40050     <031>
40495 GOTO 40050                               <161>
45060 MM=99:DIM MA$(MM),MA%(MM):MP=0           <204>
45135 LM=140:DIM LO%(LM,1):LP=0                <088>
45145 IM=270:DIM I%(IM):IP=0                   <107>
45190 SM=60:DIM S%(SM):SP=0                    <060>
45220 DI=32766                                 <186>
45250 DP=ASC(":"):KO=ASC("'"):LA=ASC("£"):
      NU$=CHR$(0):BL=ASC(" ")                  <109>
45253 BE=ASC("!"):TE=34:GT$=CHR$(137)          <067>
45254 IC$=CHR$(139):TH=167:NO$=CHR$(168):K
      M=44                                     <098>
45265 BM=13:DIM BE$(BM)                        <208>
45270 FOR I=0 TO BM:READ BE$(I):NEXT           <032>
45271 BE$(3)=IC$                               <132>
45272 DATA LOOP,EXIT,ELOOP,IF,ELSE,FI          <044>
45273 DATA CASEOF,OF,ECASE,EXT                 <133>
45274 DATA WHILE,EWHILE,REPEAT,UNTIL           <085>
45410 DEF FN AD(X)=PEEK(X)+256*PEEK(X+1)       <198>
45480 EM=40:DIM ER%(EM,1):EP=0:DIM ER$(40)     <180>
45500 FOR I=0 TO 9:READ ER$(I):NEXT            <193>
45510 DATA"FALSCHER BEFEHL","BLOCKSCHACHTE
      LUNG: ANFANG FEHLT"                      <007>
45511 DATA"UNDEFINIERTE MARKE","STACK VOLL
      "                                        <017>
45512 DATA"ZU VIELE IF/ELSE/CASE/OF","ZU V
      IELE LOOP/WHILE/REPEAT"                  <163>
45513 DATA"ZU VIELE MARKEN",,"BLOCK NICHT
      GESCHLOSSEN"                             <230>
45514 DATA"EXTERN DECLARATION"                 <113>
45600 I=0:READ W                               <247>
45610 POKE 704+I,W:I=I+1:READ W:IF W<256 T
      HEN 45610                                <154>
45620 DATA 32,115,0,8,201,33,240,4,40,76,2
      31,167                                   <139>
45630 DATA 169,8,133,44,169,138,76,231,167
      ,999                                     <127>
45650 FOR I=0 TO 10:READ W:POKE 750+I,W        <241>
45660 NEXT                                     <203>
45670 SYS 750                                  <087>
45680 DATA 169,192,141,8,3,169,2,141,9,3,9
      6                                        <098>
45999 RETURN                                   <080>
48050 IF MP=0 THEN RETURN                      <137>
48055 H=0                                      <231>
48057 PRINT"{CLR,DOWN,6SPACE}** MARKENTABE
      LLE AUSGEBEN **"                         <134>
48060 INPUT"{2DOWN,SPACE}AUF DRUCKER (J/N)
      ";B$                                     <050>
48070 IF NOT(B$="J")THEN 48091                 <036>
48075 PRINT" DRUCKER AN?":GOSUB 49550          <176>
48080 OPEN 1,4                                 <150>
48090 GOTO 48104                               <231>
48091 :                                        <058>
48100 OPEN 1,3                                 <138>
48102 H=-1                                     <243>
48104 :                                        <071>
48120 FOR I=0 TO MP-1                          <042>
48140 PRINT#1,MA%(I)+DI,MA$(I)                 <147>
48150 IF I-INT(I/10)*10=0 THEN IF I AND H
      THEN GOSUB 49550                         <097>
48180 NEXT                                     <183>
48185 CLOSE 1:GOSUB 49550                      <025>
48190 RETURN                                   <241>
49050 IF EP=0 THEN RETURN                      <057>
49055 H=0                                      <215>
49057 PRINT"{CLR,DOWN,6SPACE}** FEHLERTABE
      LLE AUSGEBEN **"                         <216>
49060 INPUT"{2DOWN,SPACE}AUF DRUCKER (J/N)
      ";B$                                     <034>
49070 IF NOT(B$="J")THEN 49091                 <084>
49075 PRINT"{DOWN,SPACE}DRUCKER AN?{2DOWN}
      ":GOSUB 49550                            <094>
49080 OPEN 1,4                                 <134>
49090 GOTO 49104                               <231>
49091 :                                        <042>
49100 OPEN 1,3                                 <122>
49102 H=-1                                     <227>
49104 :                                        <055>
49110 PRINT#1,EP;" ERRORS"                     <005>
49120 FOR I=0 TO EP-1                          <022>
49140 PRINT#1,ER%(I,0)+DI;ER$(ER%(I,1))        <158>
49150 IF I-INT(I/10)*10=0 THEN IF I AND H
      THEN GOSUB 49550                         <079>
49180 NEXT                                     <167>
49185 CLOSE 1                                  <173>
49191 GOSUB 49550                              <029>
49190 RETURN                                   <225>
49550 PRINT"->{2LEFT}";                        <238>
49560 GET B$:IF B$=""THEN 49560                <223>
49570 RETURN                                   <095>
50000 PRINT"{DOWN}* FEHLER BEHEBEN, DANN N
      EU VERSUCHEN *"                          <165>
50008 PRINT:PRINT ER$(ER);" IN ";FN AD(ZA+
      2)                                       <026>
50010 PRINT#1,CHR$(0);CHR$(0);                 <157>
50020 CLOSE 1                                  <246>
50030 GOSUB 49550                              <106>
50040 GOSUB 49050                              <212>
50050 RUN                                      <051>
```


Listing 1. Das Objektprogramm Strubs (Schluß)

```
0 '======== DIE ZEILE 1 RICHTET DIE POINTER AUF DAS BASIC-ENDE AUS. =======
1 POKE 45,PEEK(174) : POKE 46, PEEK(175) : CLR : GOTO £HPTPRG
2 '........... AB ZEILE 10 STEHEN DIE MENUEDATEN .................
5 REM"DATAZEILEN_ENTHALTEN_<PROGRAMMNAME>,_<LADENAME>
98 DATA "ENDE..........................", "................."
100 DATA @,@
200 '
1000 £MENUE
1030 :          !IF AN/10-INT(AN/10) = 0 THEN
1040 :              M'ASKEN'D'URCHLAEUFE' = AN/10-1
1050 :          !ELSE
1060 :              MD = INT(AN/10)
1070 :          !FI
1080 '.......HAUPTSCHLEIFE.........
1090 :          M'ASKENZAEHLER' = 0
1100 :          !LOOP
1110 :              GOSUB £MASKE
1120 :              T'ASTENZAEHLER' = 1
1130 :              !LOOP
1140 :                  GOSUB £PFEIL : GOSUB £TASTE
1150 :                  !CASEOF TA'STE' = 0  THEN
1160 :                          !EXIT
1170 :                      !OF TA'STE' = -1 THEN
1180 :                          T = T-1 : IF T < 1 THEN !EXIT
1190 :                      !OF TA'STE' =  1 THEN
1200 :                          T = T+1 : IF T > 10 OR T+M*10 > AN THEN !EXIT
1230 :                  !ECASE
1240 :              !ELOOP
1250 :              !CASEOF TA'STE' =  0 THEN
1260 :                      !EXIT
1270 :                  !OF TA'STE' = -1 THEN
1280 :                      M = M-1 : IF M < 0  THEN M = 0
1290 :                  !OF TA'STE' =  1 THEN
1300 :                      M = M+1 : IF M > MD THEN M = MD
1310 :              !ECASE
```

Listing 2. »Menue«, das erste Strubs-Listing

```
1320 :        !ELOOP
1325 '.......HAUPTSCHLEIFE ENDE ..
1330 GOTO £LADEN
1340 '
1500 £MASKE
1510 :        GOSUB £ZEILE1
1520 :        I = M*10+1 ' M IST MASKENSEITENZAEHLER
1530 :        !WHILE I <= M*10+10 AND I <= AN'ZAHL' !DO
1540 :           PRINT : PRINT TAB(5) P'RG'N'AME'$(I)
1550 :           I = I+1
1560 :        !EWHILE
1570 :        GOSUB £HELP
1580 RETURN
1590 '
1700 £ZEILE1
1710 :  PRINT CHR$(147)TAB(12)"PROGRAMMAUSWAHL"
1720 :  PRINT          TAB(12)"==============="
1730 RETURN
1740 '
1800 £HELP
1810 :         Z = 24 : S = 2 : GOSUB £CRSRPO.
1820 :         PRINT RO$"F1 => OBEN"RF$" ";
1823 :         PRINT RO$"F3 => UNTEN"RF$" ";
1826 :         PRINT RO$"RETURN => LOAD"RF$;
1830 RETURN
1840 '
1900 £CRSRPO.
1910 :          PRINT CHR$(19) LEFT$(CD$,Z-1) LEFT$(CR$,S-1);
1920 RETURN
1930
2000 £PFEIL
2005 '         Q IST DER VEKTOR ZUM LOESCHEN ODER SETZEN DES PFEILS
2010 '
2020 :         Z = T*2+2 : S = 2 : GOSUB £CRSRPO.
2030 :         IF Q = 1 THEN £LBL1
2040 :         PRINT "==>" : GOTO £LBL2
2050 : £LBL1   PRINT "   " : Q = 0
2060 : £LBL2
2070 RETURN
2072 '
2200 £TASTE
2210 :         !LOOP
2220 :             GET TA'STE'$ : IF TA'STE'$ = "" THEN £THIS
2230 :             !CASEOF TA'STE'$ = CHR$(133) THEN 'F1-TASTE GEDR.
2240 :                 Q = 1 : GOSUB £PFEIL
2250 :                 TA'STE' = -1 : !EXIT
2260 :             !OF     TA'STE'$ = CHR$( 13) THEN 'RETURN   GEDR.
2270 :                 TA'STE' =  0 : !EXIT
2280 :             !OF     TA'STE'$ = CHR$(134) THEN 'F3-TASTE GEDR.
2290 :                 Q = 1 : GOSUB £PFEIL
2300 :                 TA'STE' =  1 : !EXIT
2310 :             !ECASE
2320 :         !ELOOP
2330 RETURN
2332 '
2400 £LADEN
2410 :      NU'MMER' = M'ASKENZAEHLER' * 10 + T'ASTENZAEHLER
2420 :      Z'EILE' = T*2+2 : S'PALTE' = 6 : GOSUB £CRSRPO.
2430 :      PRINT RO$ P'RG'N'AME'$(NU) RF$
2440 :      IF LEFT$(P'RG'N'AME'$(NU),4) = "ENDE" THEN £SCHLUSS
2450 '      (NOCH ZU IMPLEMENTIERENDE FUNKTION: LADEN VON MPGS)
2460 '
2470 :      LOAD L'ADE'N'AME'$(NU),8
2490 'ENDE
2495 '
2700 £SCHLUSS
2710 :      PRINT CHR$(147);
2720 :      PRINT LEFT$(CD$,10) TAB(5);
2730 :      PRINT        RO$ "    F5 => MENUE      "
2740 :      PRINT TAB(5) RO$ "    F7 => ANDERE DISK"
2750 :      PRINT TAB(5) RO$ "RETURN => ENDE       "
2760 :      !LOOP
2770 :          GET TA'STE'$ : IF TA'STE'$ = "" THEN £THIS
2780 :          !CASEOF TA'STE'$ = CHR$(135) THEN 'F5-TASTE GEDR.
2790 :              RUN10000
2800 :          !OF     TA'STE'$ = CHR$(136) THEN 'F7-TASTE GEDR.
2810 :              PRINT : PRINT TAB(5) "BITTE ANDERE DISK EINLEGEN,"
2820 :                      PRINT TAB(5) "DANN BEL. TASTE DRUECKEN!"
2830 :              GET TA'STE'$ : IF TA'STE'$ = "" THEN £THIS
2840 :              OPEN15,8,15,"I" : CLOSE15
2850 :              LOAD "MENUE",8
2860 :          !OF     TA'STE'$ = CHR$( 13) THEN 'RETURN   GEDR.
2870 :              PRINT CHR$(147)
2880 :              PRINT "BITTE DISKETTE DEM LAUFWERK ENTNEHMEN,"
2890 :              PRINT "DANN LAUFWERK UND RECHNER ABSCHALTEN."
2900 :              PRINT : PRINT "BY, BY!"
2910 :              END
2920 :          !ECASE
2930 :      !ELOOP
2940 '
10000 £HPTPRG
10010 '      ******** HAUPTPROGRAMM ***********
10012 REM"_____MENUE__VERSION__VOM__01.05.84__
10014 REM"__________GERD____SOMBETZKI__________
10016 REM"__________FUHRMANNSTR.___47__________
10018 REM"__________4600_DORTMUND__13__________
10020 REM"__________TEL.:_0231/213656__________
10022 '
10030 :             GOSUB £INIT
10040 :              GOTO £MENUE
10050 '
10060 '      ***********************************
10070 '
10100 £INIT
10130 :        GOSUB £ANZAHL : IF AN'ZAHL' = 0 THEN £SCHLUSS
10140 :        DIM P'RG'N'AME'$(AN'ZAHL'), L'ADE'N'AME'$(AN'ZAHL')
10150 :        RESTORE
10160 :        FOR I = 1 TO AN'ZAHL
10170 :           READ P'RG'N'AME'$(I), L'ADE'N'AME'$(I)
10180 :        NEXT
10190 '
10200 '------------GLOBALVARIABLENDEFINITION-----------------
10210 '
10220 :        CD$ = "[illegible]" 'CURSOR DOWN'  : RO$ = "[illegible]" 'RVS ON
10230 :        CR$ = "[illegible]" 'CURSOR RIGHT' : RF$ = "[illegible]" 'RVS OFF
10235 '
10240 :        FOR I = 1 TO 7
10250 :           CD$ = CD$+CD$ : CR$ = CR$+CR$
10260 :        NEXT
10290 RETURN
10295 '
10300 £ANZAHL
10310 :        RESTORE : AN'ZAHL' = -1
10320 :        !REPEAT
10330 :            READ P'RG'N'AME'$, L'ADE'N'AME'$
10340 :            AN'ZAHL' = AN'ZAHL' + 1
10350 :        !UNTIL P'RG'N'AME'$ = "@"
10360 RETURN
10370 '

READY.
```

Listing 2. Quellcodeprogramm »Menue« (Schluß)

```
1 POKE45,PEEK(174):POKE46,PEEK(175):CLR:GOTO10000
5 REM"DATAZEILEN_ENTHALTEN_<PROGRAMMNAME>,_<LADENAME>
98 DATA"ENDE........................","............
100 DATA@,@
1000 :
1030 IFNOT(AN/10-INT(AN/10)=0)THEN1051
1040 MD=AN/10-1
1050 GOTO1070
1051 :
1060 MD=INT(AN/10)
1070 :
1090 M=0
1100 :
1110 GOSUB1500
1120 T=1
1130 :
1140 GOSUB2000:GOSUB2200
1150 IFNOT(TA=0)THEN1171
1160 GOTO1241
1170 GOTO1230
1171 IFNOT(TA=-1)THEN1191
1180 T=T-1:IFT<1THEN1241
1190 GOTO1230
1191 IFNOT(TA=1)THEN1230
1200 T=T+1:IFT>10ORT+M*10>ANTHEN1241
1230 :
1240 GOTO1130
1241 :
1250 IFNOT(TA=0)THEN1271
1260 GOTO1321
1270 GOTO1310
1271 IFNOT(TA=-1)THEN1291
1280 M=M-1:IFM<0THENM=0
1290 GOTO1310
1291 IFNOT(TA=1)THEN1310
1300 M=M+1:IFM>MDTHENM=MD
1310 :
1320 GOTO1100
1321 :
1330 GOTO2400
1500 :
1510 GOSUB1700
1520 I=M*10+1
1530 IFNOT(I<=M*10+10ANDI<=AN)THEN1561
1540 PRINT:PRINTTAB(5)PN$(I)
1550 I=I+1
1560 GOTO1530
1561 :
1570 GOSUB1800
1580 RETURN
1700 :
1710 PRINTCHR$(147)TAB(12)"PROGRAMMAUSWAHL"
1720 PRINTTAB(12)"==============="
1730 RETURN
1800 :
1810 Z=24:S=2:GOSUB1900
1820 PRINTRO$"F1 => OBEN"RF$" ";
1823 PRINTRO$"F3 => UNTEN"RF$" ";
1826 PRINTRO$"RETURN => LOAD"RF$;
1830 RETURN
1900 :
1910 PRINTCHR$(19)LEFT$(CD$,Z-1)LEFT$(CR$,S-1);
1920 RETURN
2000 :
2020 Z=T*2+2:S=2:GOSUB1900
2030 IFQ=1THEN2050
2040 PRINT"==>":GOTO2060
2050 PRINT"   ":Q=0
2060 :
2070 RETURN
2200 :
2210 :
2220 GETTA$:IFTA$=""THEN2220
2230 IFNOT(TA$=CHR$(133))THEN2261
2240 Q=1:GOSUB2000
2250 TA=-1:GOTO2321
2260 GOTO2310
2261 IFNOT(TA$=CHR$(13))THEN2281
2270 TA=0:GOTO2321
2280 GOTO2310
2281 IFNOT(TA$=CHR$(134))THEN2310
2290 Q=1:GOSUB2000
2300 TA=1:GOTO2321
2310 :
2320 GOTO2210
2321 :
2330 RETURN
2400 :
2410 NU=M*10+T
2420 Z=T*2+2:S=6:GOSUB1900
2430 PRINTRO$PN$(NU)RF$
2440 IFLEFT$(PN$(NU),4)="ENDE"THEN2700
2470 LOADLN$(NU),8
2700 :
2710 PRINTCHR$(147);
2720 PRINTLEFT$(CD$,10)TAB(5);
```

Listing 3. Das Objektprogramm »Menue«

```
2730 PRINTRO$"      F5 => MENUE          "
2740 PRINTTAB(5)RO$"      F7 => ANDERE DISK"
2750 PRINTTAB(5)RO$"RETURN => ENDE           "
2760 :
2770 GETTA$:IFTA$=""THEN2770
2780 IFNOT(TA$=CHR$(135))THEN2801
2790 RUN10000
2800 GOTO2920
2801 IFNOT(TA$=CHR$(136))THEN2861
2810 PRINT:PRINTTAB(5)"BITTE ANDERE DISK EINLEGEN,"
2820 PRINTTAB(5)"DANN BEL. TASTE DRUECKEN!"
2830 GETTA$:IFTA$=""THEN2830
2840 OPEN15,8,15,"I":CLOSE15
2850 LOAD"MENUE",8
2860 GOTO2920
2861 IFNOT(TA$=CHR$(13))THEN2920
2870 PRINTCHR$(147)
2880 PRINT"BITTE DISKETTE DEM LAUFWERK ENTNEHMEN,"
2890 PRINT"DANN LAUFWERK UND RECHNER ABSCHALTEN."
2900 PRINT:PRINT"BY, BY!"
2910 END
2920 :
2930 GOTO2760
2931 :
10000 :
10012 REM"______MENUE__VERSION__VOM__01.05.84__
10014 REM"___________GERD____SOMBETZKI___________
10016 REM"__________FUHRMANNSTR.___47____________
10018 REM"__________4600_DORTMUND__13____________
10020 REM"__________TEL.:_0231/213656____________
10030 GOSUB10100
10040 GOTO1000
10100 :
10130 GOSUB10300:IFAN=0THEN2700
10140 DIMPN$(AN),LN$(AN)
10150 RESTORE
10160 FORI=1TOAN
10170 READPN$(I),LN$(I)
10180 NEXT
10220 CD$="[illegible]":RO$="[illegible]"
10230 CR$="[illegible]":RF$="[illegible]"
10240 FORI=1TO7
10250 CD$=CD$+CD$:CR$=CR$+CR$
10260 NEXT
10290 RETURN
10300 :
10310 RESTORE:AN=-1
10320 :
10330 READPN$,LN$
10340 AN=AN+1
10350 IFNOT(PN$="@")THEN10320
10360 RETURN
READY.
```

Listing 3. Das Objektprogramm »Menue«, so wie wie es von Strubs generiert wurde (Schluß)

Wie spät ist es bitte?

Haben Sie auch schon beim Programmieren jedes Zeitgefühl verloren und vielleicht einen Termin verpaßt? Mit diesem Uhr-Programm dürfte das nicht mehr passieren. Zusätzlich zur ständigen Zeitanzeige kann noch eine Alarmfunktion aufgerufen werden.

Mit »Piep Piep Piep« erinnert Sie dieses Programm daran, daß es schon wieder nach Mitternacht ist. Es ist vielleicht besser, die neueste Version zu speichern und ins Bett zu gehen. Aber genug der Vorrede, gehen wir »in medias res«.

Das Programm CIA-Uhr wird bei jedem Interrupt des C 64 einmal abgearbeitet, also 60mal pro Sekunde. Dabei wird die Echtzeituhr des C 64 abgefragt, die Alarmzeit überprüft und die Uhrzeit auf den Bildschirm gedruckt. Im Gegensatz zu TI oder TI$ hat die Echtzeituhr eine große Ganggenauigkeit. Die Variable TI wird nämlich nur bei jedem Interrupt hochgezählt. Ein Interrupt muß aber nicht jede 60stel Sekunde stattfinden. Beispielsweise werden beim Speichern oder Laden von Programmen wesentlich weniger Interrupts pro Sekunde vom Computer ausgelöst als im READY-Modus.

Das Programm muß mit dem MSE eingegeben werden. Nach dem Laden mit LOAD"UHR",8,1 (1,1) muß NEW eingegeben werden, damit kein OUT OF MEMORY ERROR auftritt. Gestellt wird die Uhr durch SYS 49152,"Uhrzeit","Alarmzeit". Die Zeiten müssen 6stellig angegeben werden. Die Alarmzeit kann auch weggelassen werden.

Bei RUN/STOP-RESTORE verschwindet die Anzeige vom Bildschirm, während die Uhr intern aber weiterläuft. Mit SYS 49152 wird die Anzeige wieder aktiviert. Das Maschinen-Programm kann auch in andere Speicherbereiche verschoben werden. Man sollte dazu wissen, daß in den Bereichen $C0A3 bis $C0A6 und $C137 bis $C14D Tabellen stehen.

(Jörg Dorchain/hm)

```
programm : uhr                c000 c230
------------------------------------------
c000 : 78 a9 36 8d 14 03 a9 c0   0e
c008 : 8d 15 03 58 ad 0f dc 29   05
c010 : 7f 8d 0f dc a9 85 8d 0d   cc
c018 : dc ad 0e dc 09 80 8d 0e   d1
c020 : dc 4c eb c1 ad 0d dc 29   5e
c028 : 04 f0 03 20 bb c0 ad 43   68
c030 : c1 f0 03 20 c1 c0 a0 00   d3
c038 : ad 0b dc ae 08 dc 29 10   a4
c040 : 18 4a 4a 4a 4a 20 9b c0   ef
c048 : ad 0b dc ae 08 dc 29 0f   b2
c050 : a2 00 20 9b c0 bd a3 c0   78
c058 : 20 9b c0 e8 ad 0a dc 29   84
c060 : f0 4a 4a 4a 4a 20 9b c0   e7
c068 : ad 0a dc 29 0f 20 9b c0   58
c070 : bd a3 c0 20 9b c0 e8 ad   f2
c078 : 09 dc 29 f0 4a 4a 4a 4a   0c
c080 : 20 9b c0 ad 09 dc 29 0f   8e
c088 : 20 9b c0 bd a3 c0 20 9b   56
c090 : c0 e8 ad 08 dc 20 9b c0   ef
c098 : 4c 31 ea 18 69 b0 99 1e   f9
c0a0 : 04 c8 60 0a fe 0b a9 20   91
c0a8 : 8d 04 d4 a9 07 8d 05 d4   3c
c0b0 : a9 f4 8d 06 d4 a9 0f 8d   e9
c0b8 : 18 d4 60 a9 09 8d 43 c1   15
c0c0 : 60 6c 37 c1 a9 14 8d 00   ce
c0c8 : d4 a9 1a 8d 01 d4 a9 21   49
c0d0 : 8d 04 d4 a0 14 4c 10 c1   10
c0d8 : a9 20 8d 04 d4 a0 0a 4c   88
c0e0 : 10 c1 a9 cf 8d 00 d4 a9   b5
c0e8 : 22 8d 01 d4 a9 21 8d 04   8e
c0f0 : d4 a0 14 4c 10 c1 a9 20   99
c0f8 : 8d 04 d4 a0 0a 4c 10 c1   97
c100 : a0 00 8c 41 c1 a9 c4 8d   83
c108 : 37 c1 a9 c0 8d 38 c1 60   05
c110 : 8c 42 c1 a9 1d 8d 37 c1   01
c118 : a9 c1 8d 38 c1 ce 42 c1   2b
c120 : d0 14 ac 41 c1 b9 39 c1   a0
c128 : 8d 37 c1 c8 b9 39 c1 8d   62
c130 : 38 c1 c8 8c 41 c1 60 c4   3a
c138 : c0 d8 c0 e2 c0 f3 c0 00   9f
c140 : c1 00 00 00 10 00 00 00   03
c148 : 00 00 00 00 00 00 20 a6   16
c150 : c0 20 06 e2 20 fd ae 20   eb
c158 : 9e ad 20 8f ad 20 a3 b6   9f
c160 : 86 14 84 15 c9 06 f0 0a   59
c168 : a9 00 aa a8 18 d8 b8 4c   95
c170 : 48 b2 a0 ff c8 b1 14 38   14
c178 : e9 30 90 ec 99 45 c1 c0   87
c180 : 05 d0 f1 a0 ff c8 b1 14   b3
c188 : c9 3a b0 dc c0 05 d0 f5   99
c190 : a0 ff a2 00 c8 b9 45 c1   cb
c198 : 0a 0a 0a 0a 8d 44 c1 c8   ff
c1a0 : b9 45 c1 0d 44 c1 9d 4b   6d
c1a8 : c1 e8 e0 03 d0 e6 a2 00   45
c1b0 : a0 0b f8 bd 4b c1 f0 b0   b3
c1b8 : c9 24 d0 03 a9 00 18 b0   84
c1c0 : a7 c9 12 90 04 e9 12 09   cc
c1c8 : 80 99 00 dc e8 88 bd 4b   11
c1d0 : c1 c9 60 b0 93 99 00 dc   64
c1d8 : e8 88 bd 4b c1 c9 60 b0   2a
c1e0 : 87 99 00 dc d8 a9 00 8d   c5
c1e8 : 08 dc 60 20 4e c1 20 0c   06
c1f0 : c2 ad 0f dc 09 80 8d 0f   d1
c1f8 : dc 20 51 c1 20 0c c2 78   cf
c200 : a9 24 8d 14 03 a9 c0 8d   3d
c208 : 15 03 58 60 a0 ff c8 b9   62
c210 : 45 c1 f0 fa c0 09 d0 03   6f
c218 : 4c 0b e2 a0 ff a9 00 c8   95
c220 : 99 45 c1 c0 08 d0 f8 60   90
c228 : a9 00 8d 43 c1 4c a6 c0   38
```

Listing zur CIA-Uhr. Es muß mit dem MSE (siehe Seite 18) eingegeben werden.

Tiny-Forth-Compiler zum Abtippen

Die Programmiersprache Forth ist zur Zeit in aller Munde. Hier haben Sie die Möglichkeit, Forth einmal praktisch zu erleben.

Forth ist eine der jüngsten Programmiersprachen. Sie wurde 1969 von Charles Moore am National Radio Astronomy Observatory in den USA entwickelt. Der Name der Sprache lautete eigentlich Fourth (das Vierte), aber der IBM-Computer, auf dem Forth entwickelt wurde, ließ nur fünf Buchstaben als Namensangabe zu, so entstand »Forth«.

Forth ist eine der schnellsten Programmiersprachen, die es gibt. Vor allem auf Heimcomputern wird es deshalb gerne eingesetzt. Dazu kommt, daß Forth nicht viel Speicherplatz beansprucht. Die Sprache besteht nicht nur aus einem Compiler, sondern auch aus einem Interpreter; beide arbeiten Hand in Hand.

Die wohl auffälligste Eigenart von Forth ist die Art und Weise, in der Forth rechnet. Es ist die sogenannte »UPN« (Umgekehrte Polnische Notation), auch Postfix-Notation genannt. Sie ist es unter anderem, die Forth die Geschwindigkeit verleiht (10 bis 20 mal so schnell wie Basic). Doch was bedeutet UPN?

Ein Beispiel: Sie wollen das Ergebnis von 8 + 5 auf dem Bildschirm ausgeben. In Basic sähe das dann so aus: »PRINT 8 + 5«. In Forth schreibt sich das etwas anders: »5 8 + .«. Scheinbar verwirrend, aber nur auf den ersten Blick. Denn das Prinzip ist einfach. Im Mittelpunkt von Forth steht der Stack (Stapel). Man stelle sich einen Stapel Papier vor, auf den man Blätter obenauf legen kann und auch nur von oben wieder nehmen kann. Das bedeutet, das Blatt, welches Sie zuletzt draufgelegt haben, wird als erstes wieder heruntergenommen. Man nennt dieses System auch »LIFO« (Last In — First Out). Doch wie kann man damit rechnen? Kommen wir wieder zu unserem Beispiel zurück. Der Computer legt als erstes die Zahl 5 auf den Stack. Der TOS (Top of Stack) hat jetzt den Wert 5. Dann folgt die »8« nach dem gleichen Verfahren. Darauf addiert der Computer die zwei obersten Zahlen und legt das Ergebnis auf den Stack, dafür ist »+« verantwortlich. Jetzt haben wir zwar das Ergebnis auf dem Stack, können es aber nicht sehen. Für die Ausgabe von 16 Bitzahlen ist der Befehl ».« zuständig. Damit wird immer der jeweilige Wert des TOS ausgegeben.

Diese Art des Rechnens mittels UPN mag für Menschen sehr gewöhnungsbedürftig sein, für den Computer ist sie ideal. Doch Forth besitzt neben seiner Geschwindigkeit auch noch weitere Vorteile:

— In Forth können Sie Ihre eigenen Befehle definieren. Es gibt dann in der Benutzung keinen Unterschied mehr zwischen den vordefinierten und Ihren eigenen Befehlen.
— Forth besteht nicht nur aus einem Compiler, sondern auch aus einem Interpreter. Dies gibt Ihnen die Möglichkeit, selbstentwickelte Befehle sofort zu testen.
— Eine sehr hohe Geschwindigkeit beim Compilieren (1 Pass-Compiler)
— Strukturierte Programmierung ohne GOTO

Sie sehen also, Forth ist eine Sprache, mit der zu beschäftigen sich lohnt.

Die folgende Anleitung zur Handhabung des Forth-Compilers kann natürlich kein Lehrbuch ersetzen. Falls Sie mit Forth noch keinerlei Erfahrungen haben, finden Sie im Anhang eine Übersicht über Forth-Literatur. Doch nun zur Beschreibung unseres Tiny-Forth-Compilers.

Die folgenden Befehle haben so gut wie alle einen Einfluß auf den Stack. Deshalb wird eine verkürzte Schreibweise verwendet, um das Verhalten der einzelnen Befehle darzustellen: »(Stack vorher - - - Stack nachher) «.

Die verwendeten Symbole haben folgende Bedeutung: n = 16-Bit-Zahl, b = 8-Bit-Zahl, c = ASCII, addr = Adresse, f = Flag (0/1).

Die einzelnen Befehle des Tiny-Forth-Compilers sind in Tabelle 1 noch einmal übersichtlich dargestellt. Im folgenden werden die Befehle genauer beschrieben:

Arithmetikbefehle

Addition »+« (n2 n1 - - - n3)
Der Additionsbefehl holt die ersten beiden Argumente (n1,n2) vom Stack, addiert sie miteinander und legt das Ergebnis (n3) in den TOS (Top of Stack).

Subtraktion »-« (n2 n1 - - - n3)
Der Subtraktionsbefehl holt, wie bei der Addition, die ersten zwei Argumente (n1,n2) vom Stack, das Ergebnis (n3) wird wieder im TOS abgelegt.

Multiplikation »*« (n2 n1 - - - n3)
Die Multiplikation verhält sich analog zur Addition und Subtraktion.

Division »/« (n2 n1 - - - n3)
Analog zu Multiplikation.

Modulo »MOD« (n2 n1 - - - n3)
Ähnlich einer Division, es wird aber nicht das Ergebnis der Division, sondern der Divisionsrest auf den Stack gelegt.

Vergleichsbefehle

Gleich »=« (n2 n1 - - - f1)
Es werden die zwei obersten Werte auf Gleichheit geprüft.

Größer »<« (n2 n1 - - - f1)
Es wird geprüft, ob n2 größer als n1 ist.

Kleiner »>« (n2 n1 - - - f1)
Es wird geprüft, ob n2 kleiner als n1 ist.

Logische Verknüpfungen

AND (n2 n1 - - - n3)
Zwischen den Werten n1 und n2 wird eine logische UND-Operation ausgeführt.

OR = (n2 n1 - - - n3)
Es wird ein logisches ODER ausgeführt.

XOR (n2 n1 - - - n3)
Es wird ein logisches exklusives ODER ausgeführt.

NOT (f1 - - - f2)
Das oben liegende Flag wird invertiert.

Stackoperatoren

DROP (n2 n1 - - - n2)
Der oberste Wert wird vom Stack entfernt.

DUP (n1 - - - n1 n1)
Der oberste Wert auf dem Stack wird dupliziert.

SWAP (n2 n1 - - - n1 n2)
Die obersten beiden Werte werden vertauscht.

OVER (n2 n1 - - - n2 n1 n2)
Kopiert den zweiten Wert zum neuen TOS.

PICK (n1 n - - - n1 n2)
Pick holt den n-ten Wert in den TOS.

ROT (n3 n2 n1 - - - n2 n1 n3)
Rot läßt die ersten drei Elemente des Stack gegen den Uhrzeigersinn rotieren.

@ (addr - - - n1)
Holt eine 16-Bit-Zahl aus der Adresse addr.

! (n1 addr - - -)
Speichert eine 16-Bit-Zahl in der Adresse addr.

c@ (addr - - - b1)

Holt eine 8-Bit-Zahl aus der Adresse addr.
c! (b1 addr - - -)
Speichert eine 8-Bit-Zahl in der Adresse addr.

Kontrollstrukturen

BEGIN — UNTIL (f - - -)
Der Programmteil zwischen BEGIN und UNTIL wird so lange ausgeführt, bis der TOS bei UNTIL ungleich Null ist.
BEGIN — WHILE — REPEAT (f - - -)
Der Programmteil zwischen BEGIN und REPEAT wird so lange ausgeführt, wie der TOS bei WHILE ungleich Null ist.
IF — ENDIF (f - - -)
Der Programmteil zwischen IF und ENDIF wird nur dann ausgeführt, wenn der TOS bei IF ungleich Null ist.
IF — ELSE —ENDIF (f - - -)
Bei erfüllter Bedingung wird der Programmteil zwischen IF und ELSE ausgeführt, bei nichterfüllter Bedingung der zwischen ELSE und ENDIF.

Schleifen

DO (n2 n1 - - -)
DO legt die Argumente n1 (Endwert), n2 (Startwert) auf den Returnstack und leitet eine Schleife ein.
LOOP (- - -)
LOOP erhöht den Schleifen-Zähler um 1, ist der Endwert nicht erreicht, wird wieder zu dem auf DO folgenden Befehl gesprungen.
I (- - - n)
Der Befehl I legt den Wert des Schleifenzählers auf den TOS.
+LOOP (n - - -)
+LOOP erhöht den Schleifenzähler um n, weiter wie LOOP.

Ein-/Ausgabeoperatoren

KEY (- - - c)
Holt den ASCII-Wert, der gerade gedrückten Taste in den TOS, ist keine Taste gedrückt, so wird eine 0 in den TOS gelegt.
GET (- - - c)
Wartet, bis eine Taste gedrückt wird und legt dann ihren ASCII-Wert in den TOS.
EXPECT (addr n - - -)
Erwartet eine Eingabe, die mit Return abgeschlossen wird, und legt sie bei addr mit einer maximalen Länge n im Speicher ab. Als Abschlußzeichen wird eine 13 in den Speicher gesetzt.
EMIT (c - - -)
EMIT gibt den auf dem TOS liegenden ASCII-Wert auf dem Bildschirm aus.
TYPE (addr n - - -)
TYPE gibt n Zeichen, welche ab addr im Speicher stehen, auf dem Bildschirm aus.
CR (- - -)
CR bewirkt einen Zeilenvorschub.
CLS (- - -)
Löscht den Bildschirm.
."TEXT" (- - -)
Gibt den Text zwischen ».''« und »"« aus (funktioniert nur in compilierter Form).

Definitionsbefehle

n CONSTANT ⟨Name⟩ (n - - -)
CONSTANT definiert eine Konstante mit dem Wert n und dem Namen ⟨Name⟩. Wird ⟨Name⟩ im Programm aufgerufen, so wird n auf den TOS gelegt.
n VARIABLE ⟨Name⟩ (n - - -)
VARIABLE definiert eine Variable mit dem Wert n und dem Namen ⟨Name⟩. Wird ⟨Name⟩ im Programm aufgerufen, so wird die Adresse der Varible auf den Stack gelegt. Ein Wert wird mit @ (lies: Fetch) auf den TOS geholt und mit ! (lies: Store) an eine Variable übergeben.
n MEMORY ⟨Name⟩ (n - - -)
MEMORY definiert einen Speicherbereich mit dem Namen ⟨Name⟩ und der Länge n. Wird ⟨Name⟩ im Programm aufgerufen, so wird die Adresse des Speicherbereichs übergeben. In Adresse-2 ist die Länge zu finden und kann mit @ ausgelesen werden. MEMORY ist kein Standardwort!
: ⟨Name⟩...; (- - -)
Der Doppelpunkt definiert ein neues Forth-Wort mit dem Namen ⟨Name⟩. Die Definition muß mit »;« abgeschlossen werden.

Systembefehle

BASIC: Kehrt zum Basic zurück.
RESET: Kehrt zum Ausgangszustand zurück.
LIST n: Listet Screen n auf dem Drucker oder dem Bildschirm.
CLEAR n: Löscht SCREEN n auf der Diskette.
LOAD n: Compiliert Screen n in das Vocabulary.
FORGET ⟨Name⟩: Löscht das Wort ⟨Name⟩ aus dem Vocabulary.
VLIST: Listet das Vocabulary.
SAVE-SYSTEM ⟨Name⟩: Speichert den Objektcode und Vocabulary-Einträge aller selbstdefinierten Befehle auf Diskette.
LOAD-SYSTEM ⟨Name⟩: Lädt den Objektcode und die Vocabulary-Einträge wieder. Das System muß vorher mit RESET wieder in den Ausgangszustand gebracht werden und in Zeile 380 muß die Variable VOC denselben Wert wie beim Speichern haben, da sonst die Sprungadressen im Objektcode nicht stimmen.
EDIT n: Ruft den Bildschirmeditor auf, n ist die Nummer des Screens.

Spezialbefehle

CALL (addr - - -)
Ruft ein Maschinenprogramm mit der Adresse addr auf. Es können, wie bei SYS in Basic, in den Speicherzellen 780 das A-, 781 das X-, 782 das Y-Register mit übergeben werden.
R⟩ (- - - n)
Bringt das oberste Element des Return-Stacks auf den Stack.
⟩R (n - - -)
Bringt das oberste Element des Stacks auf den Return-Stack.
;S (- - -)
Dieser Befehl sorgt für den vorzeitigen Abbruch des aktuellen Befehls.

Das Programm

Das Programm (Listing 1) wurde mit Absicht sehr flexibel gehalten. Nach oder während des Abtippens können Sie einige Dinge nach Ihrem eigenen Ermessen ändern. So zum Beispiel die Startadresse des Objektcodes in Zeile 380; die Variable VOC enthält den Startwert. Allerdings sollten Sie nicht unter VOC=5*4096 gehen, da sonst eine Kollision mit dem String-Bereich möglich ist. Dies macht sich durch eine »OUT OF MEMORY«-Meldung oder durch einen Systemabsturz bemerkbar. Das Programm kann nur mit 16-Bit-Zahlen arbeiten und umfaßt nicht den gesamten Forth-Standard. Das Programm besteht aus einem Compiler, der in den Zeilen 1540 bis 2610, und einem Interpreter, der in den Zeilen 700 bis 1530 steht. Der Interpreter ist nicht in der Lage, alle Befehle auszuführen. Um Ihnen zu helfen, welche Befehle interpretierbar sind und welche nicht, sind nur die interpretierbaren im Vocabulary aufgeführt. Andere Befehle wie VARIABLE, MEMORY etc., kurzum alle Definitionsbefehle, dürfen nicht im Compilermodus angewendet werden.

Programmieren in Forth

Nach dem Starten des Programms wird zuerst das Maschinenprogramm »VOCABULARY« (Listing 2) in den Bereich 49152 bis 50160 geladen. Nach etwa einer Sekunde er-

scheint ein blinkender Cursor; jetzt können Sie Ihre Eingaben machen. Sie befinden sich im Interpretermodus, das heißt alle eingegebenen Befehle werden sofort ausgeführt. Daß dies im Interpretermodus so langsam geht, liegt nur daran, daß der Interpreter in Basic geschrieben ist; compilierte Befehle laufen dagegen etwa 10- bis 20mal so schnell wie Basic (sie werden vollkommen in Maschinensprache übersetzt). Vielleicht geben Sie mal das folgende Beispiel ein: »8 5 + .« (RETURN). Sie müßten jetzt 13 auf dem Bildschirm erhalten; wenn nicht, dann müssen Sie irgendwo im Programm einen Fehler gemacht haben. Sie können auch Kommentare einfügen, sie werden mit einer »(« begonnen und mit »)« abgeschlossen. Jeder Befehl wird durch ein Leerzeichen (oder Return) vom anderen getrennt.

Doch wie definiert man einen neuen Befehl? Diese Frage wird Ihnen sicher schon lange auf den Lippen brennen. Doch auch hier macht es Ihnen Forth sehr einfach. Um die Definition eines neuen Befehls einzuleiten wird »:« benutzt, gefolgt von dem Namen des neuen Befehls (bitte vergessen Sie nicht das Leerzeichen hinter jedem Befehl.) Dann folgen die Befehle, die in das Wort compiliert werden sollen. Ein »;« beendet die Definition. Danach ist der Befehl wie jeder andere Befehl benutzbar. Auf Fehler reagiert der Compiler, indem er die Compilation abbricht. Danach sollte man den Befehl mit FORGET löschen, da sonst das Programm abstürzt, wenn Sie den Befehl aufrufen.

Der Editor

Nun ist es ziemlich zeitaufwendig, wenn man bei jedem Fehler den Befehl neu eingeben muß, deshalb bietet Forth einen zweiten Editor. Es ist ein Bildschirmeditor, welcher mit dem Systembefehl »EDIT n« aufgerufen wird. n bezeichnet hier die Nummer des Screens, der editiert werden soll. Ein Screen ist einfach eine Bildschirmseite, auf der der zu compilierende Sourcecode steht.

Geben Sie einmal »EDIT 1« ein. Das System versucht nun, Screen 1 von der Diskette zu laden. Ist der Screen nicht auf Diskette vorhanden, so wird trotzdem in den Editor gesprungen, nur daß der Screen leer ist. Nachdem sich das System im Editor befindet, sehen Sie links die Zeilennummern von 0 bis 23 mit folgendem Doppelpunkt (dieser Doppelpunkt hat keine Bedeutung). Sie können nun mit dem Cursor hinter den Doppelpunkt fahren und eine Zeile eingeben. Jede Zeile muß mit RETURN abgeschlossen werden. Eine Zeile darf nicht länger als 35 Zeichen sein. Geben Sie doch einmal das vorherige Beispiel ein. Die Nummer am Anfang jeder Zeile entspricht in etwa einer Zeilennummer in Basic.

Um nun den Editor zu verlassen, gibt es zwei Möglichkeiten, einmal mit »e«, damit der Screen n abgespeichert wird und mit »s«, so wird der Editor ohne Änderung des Screens verlassen. Die Buchstaben müssen in der ersten Spalte einer Zeile stehen, also dort, wo die Zeilennummer steht. Es gibt noch mehr dieser Editorbefehle (siehe Tabelle 4). Zum Einfügen von Zeilen benützen Sie »I Zeile Anzahl«, mit »D Zeile Anzahl« löschen Sie Zeilen. Mit »L« listen Sie den Screen noch einmal, das ist dann von Nutzen, wenn Sie versehentlich die CLR-Taste betätigt haben. Mit »N Nummer« ändern Sie die Nummer eines Screens. Das Compilieren eines Screens geschieht mit LOAD n. Soll der nächste Screen (n+1) auch noch compiliert werden, so muß in der letzten Zeile des Screens n der Befehl »-- > « vorhanden sein. Experimentieren Sie doch mal ein bißchen mit dem Editor.

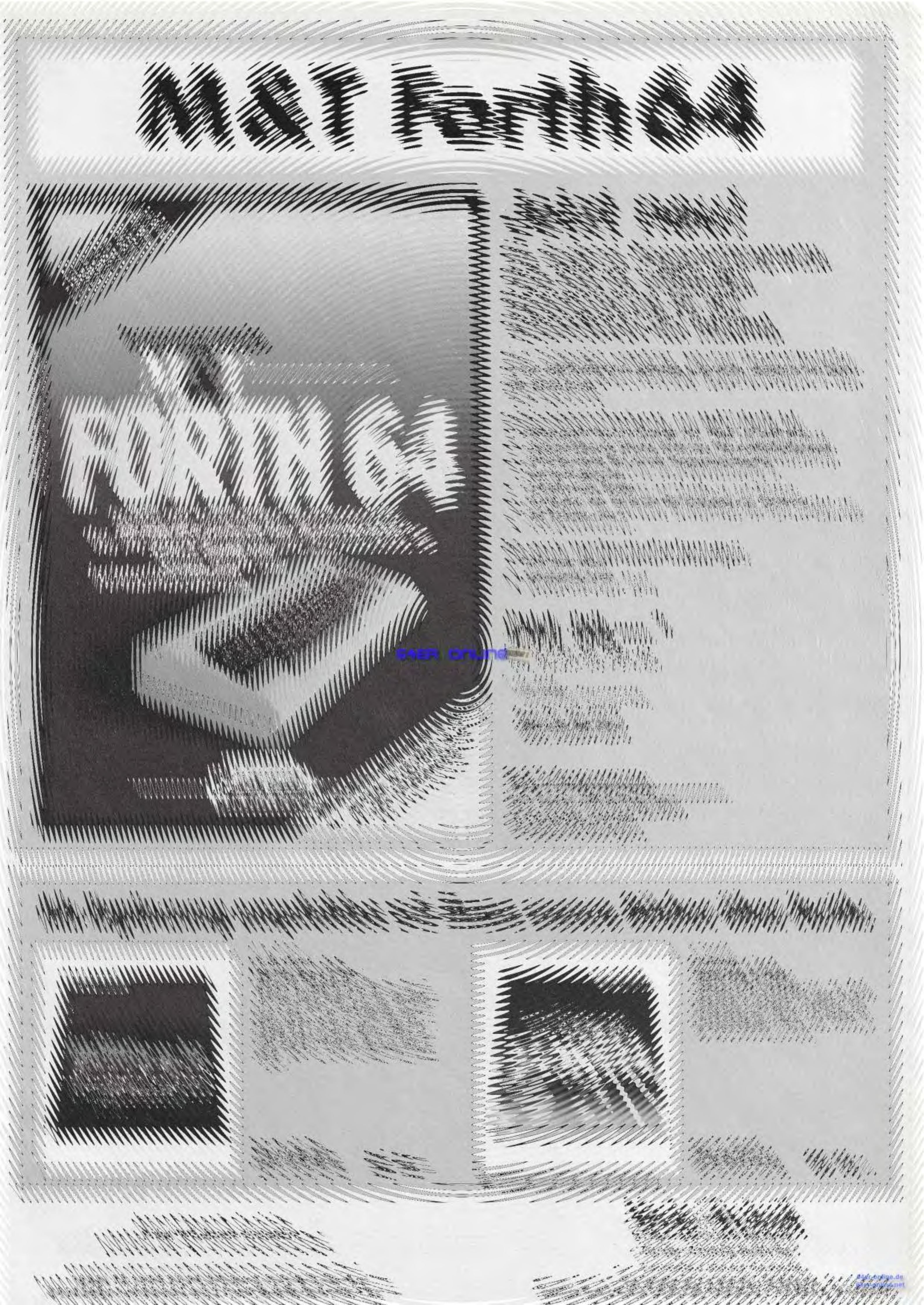
In den Listings 3 bis 7 finden Sie einige selbstdefinierte Befehle. Die Befehle J und LEAVE in Listing 4 möchte ich näher erklären. J gibt den Schleifenwert der zweitinnersten Schleife aus. Die Befehle R> und >R manipulieren den Returnstack (der Returnstack funktioniert genauso wie der normale Stack). Hier werden Werte für Schleifen zwischengespeichert und zwar im Format Endwert, Zähler. R> holt den obersten Wert des Returnstacks auf den Stack, >R tut das Gegenteil. Die 704 ist nur Zwischenspeicher. LEAVE schließt eine Schleife vorzeitig ab, indem Endwert und Zähler gleichgesetzt werden.

Viel Spaß beim Programmieren in Forth.

(Alexander Schindowski/ev)

Befehl	Stack
1) Rechenoperationen:	
1.1) +	(n2 n1 - - - n3)
1.2) -	(n2 n1 - - - n3)
1.3) *	(n2 n1 - - - n3)
1.4)/	(n2 n1 - - - n3)
1.5) MOD	(n2 n1 - - - n3)
2) Vergleichsoperatoren:	
2.1) =	(n2 n1 - - - f)
2.2) >	(n2 n1 - - - f)
2.3)<	(n2 n1 - - - f)
3) Logische Verknüpfungen:	
3.1) AND	(n2 n1 - - - n3)
3.2) OR	(n2 n1 - - - n3)
3.3) NOT	(f1 - - - f2)
3.4) XOR	(n2 n1 - - - n3)
4) Stackoperatoren:	
4.0) DROP	(n1 - - -)
4.1) DUP	(n1 - - - n1 n1)
4.2) SWAP	(n2 n1 - - - n1 n2)
4.3) OVER	(n2 n1 - - - n2 n1 n2)
4.4) PICK	(n1 n - - - n1 n2)
4.5) ROT	(n3 n2 n1 - - - n2 n1 n3)
4.6) @	(addr - - - n1)
4.7) !	(n1 addr - - -)
4.8) c@	(addr - - - b1)
4.9) c!	(b1 addr - - -)
5) Kontrollstrukturen:	
5.1) BEGIN — UNTIL	(f - - -)
5.2) BEGIN — WHILE — REPRAT	(f - - -)
5.3) IF — ENDIF	(f - - -)
5.4) IF — ELSE — ENDIF	(f - - -)
6) Definitionsworte:	
6.1) VARIABLE	(n - - -)
6.2) CONSTANT	(n - - -)
6.3) MEMORY	(n - - -)
6.4) : ... ;	
7) Ein-/Ausgabe:	
7.1) EXPECT	(addr n - - -)
7.2) TYPE	(addr n - - -)
7.3) KEY	(- - - c)
7.4) GET	(- - - c)
7.5) EMIT	(c - - -)
7.6) ."	(- - -)
7.7) .	(n - - -)
7.8) CR	(- - -)
7.9) CLS	(- - -)
8) Schleifen:	
8.1) DO	(n2 n1 - - -)
8.2) LOOP	(- - -)
8.3) +LOOP	(n - - -)
8.4) I	(- - - n)
9) Sonstige Befehle:	
9.1) call	(addr - - -)
9.2) >R	(n - - -)
9.3) R>	(- - - n)
9.4) DEPTH	(- - - n)

Tabelle 1. Der Befehlssatz des Tiny-Forth-Compilers



```
programm : vocabulary       c006 c3ea
--------------------------------------
c006 : a9 93 4c d2 ff 20 3f c0   66
c00e : 85 62 86 63 a2 90 a5 62   dc
c016 : 30 09 a9 20 20 d2 ff 38   42
c01e : 4c d4 bd a9 2d 20 d2 ff   98
c026 : 18 4c d4 bd 85 fe 86 fd   b7
c02e : a6 02 a5 fe 9d 00 cf e8   09
c036 : a5 fd 9d 00 cf e8 86 02   a4
c03e : 60 a6 02 ca bd 00 cf 85   f1
c046 : fd ca bd 00 cf 85 fe 86   4a
c04e : 02 a5 fe a6 fd 60 4c 3f   4a
c056 : c0 20 3f c0 20 2a c0 4c   fd
c05e : 2e c0 20 3f c0 8d 00 c0   d6
c066 : 8e 01 c0 20 3f c0 8d 02   dd
c06e : c0 8e 03 c0 ad 00 c0 ae   89
c076 : 01 c0 20 2a c0 ad 02 c0   28
c07e : ae 03 c0 4c 2a c0 c6 02   2f
c086 : c6 02 20 3f c0 e6 02 e6   56
c08e : 02 e6 02 e6 02 4c 2a c0   0d
c096 : a6 fc ca bd 00 ce 85 fd   ad
c09e : ca bd 00 ce 85 fe 86 fc   85
c0a6 : a5 fe a6 fd 60 85 fe 86   6f
c0ae : fd a6 fc a5 fe 9d 00 ce   6d
c0b6 : e8 a5 fd 9d 00 ce e8 86   cb
c0be : fc 60 38 ad 00 c0 ed 02   70
c0c6 : c0 8d 04 c0 ad 01 c0 ed   28
c0ce : 03 c0 8d 05 c0 60 18 ad   00
c0d6 : 00 c0 6d 02 c0 8d 04 c0   dc
c0de : ad 01 c0 6d 03 c0 8d 05   60
c0e6 : c0 60 a9 0d 4c d2 ff 20   7e
c0ee : 3f c0 8e 00 c0 8d 01 c0   2f
c0f6 : 20 3f c0 8e 02 c0 8d 03   1a
c0fe : c0 4c c0 c0 20 ed c0 ad   fc
c106 : 04 c0 0d 05 c0 f0 06 a9   4d
c10e : 00 aa 4c 2a c0 a9 00 a2   5a
c116 : 01 4c 2a c0 20 ed c0 30   b5
c11e : 06 a9 00 aa 4c 2a c0 a9   bb
c126 : 00 a2 01 4c 2a c0 20 60   2b
c12e : c0 4c 1a c1 20 3f c0 05   dc
c136 : fd f0 06 a9 00 aa 4c 2a   3d
c13e : c0 a9 00 a2 01 4c 2a c0   c4
c146 : 20 3f c0 8e 00 c0 8d 01   46
c14e : c0 20 3f c0 8e 02 c0 8d   1d
c156 : 03 c0 60 20 46 c1 ad 00   ff
c15e : c0 2d 02 c0 85 fd ad 01   4e
c166 : c0 2d 03 c0 85 fe 4c 2e   74
c16e : c0 20 46 c1 ad 00 c0 0d   00
c176 : 02 c0 85 fd ad 01 c0 0d   f9
c17e : 03 c0 85 fe 4c 2e c0 20   9c
c186 : 46 c1 ad 00 c0 4d 02 c0   18
c18e : 85 fd ad 01 c0 4d 03 c0   a1
c196 : 85 fe 4c 2e c0 20 46 c1   1d
c19e : 20 d4 c0 ae 04 c0 ad 05   35
c1a6 : c0 4c 2a c0 20 60 c0 20   77
c1ae : 46 c1 20 c0 c0 ae 04 c0   08
c1b6 : ad 05 c0 4c 2a c0 20 46   55
c1be : c1 a0 00 8c 04 c0 8c 05   e3
c1c6 : c0 a0 10 0e 04 c0 2e 05   a5
c1ce : c0 2e 02 c0 2e 03 c0 90   5d
c1d6 : 1d 18 ad 04 c0 6d 00 c0   e4
c1de : 8d 04 c0 ad 05 c0 6d 01   61
c1e6 : c0 8d 05 c0 90 08 ee 02   cf
c1ee : c0 d0 03 ee 03 c0 88 d0   af
c1f6 : d2 ae 04 c0 ad 05 c0 4c   d7
c1fe : 2a c0 20 46 c1 a0 10 a9   0e
c206 : 00 8d 04 c0 8d 05 c0 2e   46
c20e : 02 c0 2e 03 c0 2e 04 c0   6b
c216 : 2e 05 c0 38 ad 04 c0 ed   d8
c21e : 00 c0 aa ad 05 c0 ed 01   ef
c226 : c0 90 06 8d 05 c0 8e 04   fa
c22e : c0 88 d0 db 2e 02 c0 2e   34
c236 : 03 c0 60 20 00 c2 ae 02   8a
c23e : c0 ad 03 c0 4c 2a c0 20   07
c246 : 00 c2 ae 04 c0 ad 05 c0   e2
c24e : 4c 2a c0 20 96 c0 4c 2a   d8
c256 : c0 20 3f c0 4c ab c0 20   74
c25e : 60 c0 20 57 c2 4c 57 c2   83
c266 : 20 51 c2 e6 fc e6 fc 60   78
c26e : 20 51 c2 20 57 c0 20 51   8a
c276 : c2 20 60 c0 20 ed c0 30   4d
c27e : 08 20 54 c0 20 54 c0 38   dc
c286 : 60 e6 fc e6 fc 20 9b c1   38
c28e : 20 57 c2 18 60 a9 00 a2   a6
c296 : 01 20 2a c0 4c 6e c2 20   cd
c29e : 3f c0 e6 02 e6 02 a5 fe   4a
c2a6 : 10 c6 20 51 c2 20 57 c0   58
c2ae : 20 51 c2 4c 7a c2 20 3f   6e
c2b6 : c0 05 fd f0 02 38 60 18   2a
c2be : 60 20 3f c0 8a 4c d2 ff   6d
c2c6 : 20 87 ea e0 ff f0 04 e0   da
c2ce : 0d b0 02 a2 00 a9 00 4c   ee
c2d6 : 2a c0 20 c6 c2 20 57 c0   4d
c2de : 20 b4 c2 b0 06 20 54 c0   53
c2e6 : 4c d8 c2 60 86 fd 85 fe   c7
c2ee : a0 00 b1 fd f0 06 20 d2   20
c2f6 : ff c8 d0 f6 60 20 3f c0   f2
c2fe : 86 f7 20 3f c0 a0 00 b1   e4
c306 : fd 20 d2 ff c8 c4 f7 d0   fc
c30e : f6 60 20 3f c0 86 f9 20   8d
c316 : 3f c0 86 f7 85 f8 a2 00   00
c31e : a0 00 20 cf ff 91 f7 c9   c0
c326 : 0d f0 06 e8 c8 c4 f9 d0   86
c32e : f1 60 a5 02 4a aa a9 00   9a
c336 : 4c 2a c0 20 3f c0 86 f7   cf
c33e : 85 f8 20 3f c0 a0 01 91   68
c346 : f7 8a 88 91 f7 60 20 3f   58
c34e : c0 86 f7 85 f8 20 3f c0   0f
c356 : a0 00 8a 91 f7 60 20 3f   4d
c35e : c0 86 f7 85 f8 a0 00 b1   08
c366 : f7 aa c8 b1 f7 4c 2a c0   27
c36e : 20 3f c0 86 f7 85 f8 a0   ff
c376 : 00 b1 f7 aa a9 00 4c 2a   c2
c37e : c0 20 3f c0 ad 0c 03 ae   db
c386 : 0d 03 ac 0e 03 6c fd 00   8d
c38e : 20 3f c0 a4 02 8c 00 c0   19
c396 : 8a 0a 85 02 20 3f c0 ac   1f
c39e : 00 c0 84 02 4c 2a c0 20   b9
c3a6 : 3f c0 8e 00 c0 8d 01 c0   e7
c3ae : 20 3f c0 8e 02 c0 8d 03   d2
c3b6 : c0 20 3f c0 8e 04 c0 8d   95
c3be : 05 c0 ae 02 c0 ad 03 c0   16
c3c6 : 20 2a c0 ae 00 c0 ad 01   c0
c3ce : c0 20 2a c0 ae 04 c0 ad   aa
c3d6 : 05 c0 4c 2a c0 20 fd ae   f6
c3de : 20 d4 e1 a9 01 85 b9 a9   8c
c3e6 : 00 4c d5 ff 00 00 00 00   82
```

Listing 2. Maschinenspracheteil von Forth (Bitte beachten Sie die Eingabehinweise auf Seite 18)

```
 0: ( *** Zusatz-Befehle 0 *** )
 1:
 2:
 3: ( Processor-Register )
 4:
 5: 780 constant a-reg
 6: 781 constant x-reg
 7: 782 constant y-reg
 8: ( ---------------------- )
 9:
10: ( Die folgenden Befehle   )
11:
12: ( sollen zeigen,dass es   )
13:
14: ( mit dem recht beschei-  )
15:
16: ( denen Grundvokabular    )
17:
18: ( moeglich ist,doch ein   )
19:
20: ( sinnvolles,zweckmaess-  )
21:
22: ( iges zu erstellen.      )
23:-->
```

Listing 3. Zusatz-Befehle, Screen 0

```
 0:( *** zusatz-befehle 1 ***)
 1:
 2:
 3: (  2. Schleifenindex )
 4:
 5: : j
 6:     r> r> r>
 7:     dup 704 !
 8:     >r >r >r
 9:     704 @
10: ;
11:
12:
13:
14: (   Verlasse Schleife )
15:
16: : leave
17:     r> r>
18:     drop dup
19:     >r >r
20: ;
21:
22:
23: -->
```

Listing 4. Zusatz-Befehle, Screen 1

```
 0: ( *** Zusatz-Befehle 2 *** )
 1:
 2:
 3: ( open (addr 1 log nr sec --) )
 4:
 5: : open
 6:     185 c!   186 c! 184 c!
 7:     183 c!   187 !
 8:     65472 call
 9: ;
10:
11:
12:
13: ( close (log --) )
14:
15: : close
16:     cr 65484 call
17:     a-reg c!
18:     65475 call
19: ;
20:
21:
22:
23: -->
```

Listing 5. Zusatz-Befehle, Screen 2

```
 0: ( *** Zusatz-Befehele 3 *** )
 1:
 2:
 3: ( Ausgabe auf File (log --) )
 4:
 5: : out
 6:     x-reg c!
 7:     65481 call
 8: ;
 9:
10:
11:
12: ( Eingabe von File (log --) )
13:
14: : in
15:     x-reg c!
16:     65478 call
17: ;
18:
19:
20:
21:
22:
23: -->
```

Listing 6. Zusatz-Befehle, Screen 3

```
 0: ( *** Zusatz-Befehle 4 *** )
 1:
 2:
 3: ( Ausgabe auf Drucker )
 4:
 5: : printer
 6:     0 0 4 4 7 open
 7:     4 out
 8:     cr
 9: ;
10:
11:
12: ( Ausgabe auf Display )
13:
14: : display
15:     cr
16:     4 close
17: ;
18:
19:
20:
21: ( * ende * )
22:
23:
```

Listing 7. Zusatz-Befehle, Screen 4

A,X,I,Z Lauf- und Hilfsvariablen	
VOC	Vocabulary-Zeiger
BE	Beginn des Vocabulary
SP	Zeiger für Compilerstack
AN	Anzahl der Vocabulary-Einträge
ZE$	Eingabezeile
BE$	Einzelner Befehl
COMP	Flag für Compilation
AD	Adressen-Zwischenspeicher
OK	Zahlenumwandlung geglückt?
A$	Hilfsvariable
WO$(X)	Wörterverzeichnis
AD(X)	Startadressen-Verzeichnis
BLOCK	Flag für Compilation von Diskette
Z1	Zeilenzähler
XX	Umgewandelte Zahl
X	Top of Stack (TOS)

Tabelle 2. Variablenübersicht zum Tiny-Forth-Compiler

BASIC	Zurück zu Basic
RESET	Zurück in Ausgangsstellung
VLIST	Listet das Vocabulary
FORGET	Lösche ein Wort
FLOPPY	Gibt einen Befehl an die Floppy
SAVE-SYSTEM	Speichert das Vocabulary ab
LOAD-SYSTEM	Lädt das Vocabulary
LIST n	Listet eine Screen
LOAD n	Compiliert eine Screen
CLEAR n	Löscht eine Screen
EDIT n	Ruft den Editor auf

Tabelle 3. Forth-Systembefehle

E	Beendet das Editieren und speichert die Screens ab
S	Beendet das Editieren
L	Listet die Screens
I (z,a)	Fügt Zeile(n) ein
D (z,a)	Löscht Zeile(n)
N (x)	Ändert die Screen-Nummer

Tabelle 4. Befehle des Forth-Editors

Info: Literatur zu Forth:

— Paul M. Chirlien, Der Einstieg in Forth, Markt & Technik 1985, 338 Seiten, 58 Mark.
— E. Floegel, Forth-Handbuch, Ing. W. Hofacker Verlag 1983, 192 Seiten, 39 Mark.
— Monadjemi, Das Trainingsbuch zu Forth, Data Becker 1985, 300 Seiten, 39 Mark.
— Ronald Zech, Die Programmiersprache Forth, Franzis-Verlag 1984, 312 Seiten, 69 Mark.
— Weitere Literatur mit Informationen über Forth sind zu beziehen über die deutsche Sektion der Forth Interest Group (FIG). Kontaktadresse: Forth Gesellschaft Deutschland (F16), Postfach 202264, 2000 Hamburg 20. Da die FIG nicht kommerziell arbeitet, bitte bei Anfragen Freiumschlag beifügen.

```
100 REM ****************************          <091>
110 REM *                          *          <159>
120 REM *        FORTH-COMPILER    *          <034>
130 REM *                          *          <179>
140 REM *           FUER           *          <205>
150 REM *                          *          <199>
160 REM *        COMMODORE-64      *          <181>
170 REM *                          *          <219>
180 REM ****************************          <171>
190 REM *                          *          <239>
200 REM *  ALEXANDER SCHINDOWSKI    *          <251>
210 REM *                          *          <003>
220 REM *  6000 FRANKFURT/MAIN 50   *          <048>
230 REM *                          *          <023>
240 REM * RUDOLF-HILFERDING-STR.49 *          <149>
250 REM *                          *          <043>
260 REM ****************************          <253>
270 REM *                          *          <065>
280 REM *  TELEPHON: (069)/570520   *          <252>
290 REM *                          *          <085>
300 REM ****************************          <037>
310 :                                         <032>
320 :                                         <042>
330 :                                         <052>
340 IF A=0 THEN A=1:LOAD"VOCABULARY",8,1     <086>
350 DEF FN H(X)=(INT(X/256))                  <191>
360 DEF FN L(X)=(X-256*FN H(X))               <030>
370 POKE 53272,23:PRINT"{CLR,LIG.BLUE}";CH
    R$(8);                                    <131>
380 VOC=6*4096:BE=VOC:SP=0:Z1=0               <155>
390 POKE 55,FN L(BE):POKE 56,FN H(BE)         <236>
395 DIM ST(20),SC$(24),WO$(100),AD(100)       <114>
400 PRINT TAB(14);"FORTH-COMPILER"            <018>
410 PRINT TAB(17);"FUER DEN"                  <115>
420 PRINT TAB(15);"COMMODORE-64"              <086>
430 PRINT"---------------------------------
    --------";                                <156>
440 PRINT"{5SPACE}VON ALEXANDER SCHINDOWSK
    I 1985{8DOWN}"                            <092>
450 DATA 38                                   <238>
460 DATA "+",49563                            <028>
470 DATA "CLS",49158,"DEPTH",49968            <242>
480 DATA "@",50012,"DROP",49236               <176>
490 DATA "EMIT",49855,"EXPECT",49936          <206>
500 DATA "=",49410,"I",49766                  <250>
510 DATA "KEY",49880                          <164>
520 DATA "+LOOP",49821,"MOD",49733            <091>
530 DATA "NOT",49458,"OVER",49284             <027>
540 DATA ".",49163,"-",49578                  <147>
550 DATA "SWAP",49248,">R",49751              <128>
560 DATA "AND",49497,"CR",49384               <086>
570 DATA "/",49721,"DO",49757,"!",49977       <084>
580 DATA "DUP",49239,"XOR",49541              <015>
590 DATA "GET",49862,">",49434                <131>
600 DATA "<",49452,"LOOP",49811               <134>
610 DATA "*",49596,"OR",49519                 <164>
620 DATA "C@",50030,"C!",49996                <225>
630 DATA "R>",49745,"TYPE",49915              <188>
640 DATA "PICK",50062,"CALL",50047,"ROT",5
    0085                                      <056>
650 READ AN                                   <154>
660 FOR I=1 TO AN                             <080>
670 READ WO$(I),AD(I)                         <061>
680 NEXT I:POKE 2,0:POKE 252,0                <134>
690 GOSUB 3830                                <122>
693 :                                         <161>
695 REM ***************************           <227>
700 REM *** BEFEHLS-AUSWERTUNG ***            <130>
705 REM ***************************           <237>
708 :                                         <176>
710 GOSUB 2630                                <102>
715 :                                         <183>
720 IF BE$=":" THEN 1540                      <051>
725 :                                         <193>
730 FOR I=AN TO 1 STEP -1                     <033>
740 IF BE$=WO$(I) THEN 760                    <118>
750 NEXT I:GOTO 770                           <178>
760 SYS AD(I):GOTO 700                        <092>
765 :                                         <233>
770 GOSUB 3030                                <076>
780 IF OK=0 THEN 830                          <001>
790 POKE 781,FN L(XX)                         <010>
800 POKE 780,FN H(XX)                         <102>
810 SYS 49194                                 <108>
820 GOTO 700                                  <034>
825 :                                         <039>
830 IF BE$="RESET" THEN RUN                   <035>
835 :                                         <049>
840 IF BE$="BASIC" THEN END                   <031>
845 :                                         <059>
850 IF BE$<>"VLIST" THEN 900                  <240>
860 PRINT:FOR I=AN TO 1 STEP-1                <174>
870 PRINT WO$(I)"{2SPACE}";                   <128>
880 NEXT:PRINT                                <234>
890 GOTO 700                                  <104>
895 :                                         <109>
900 IF BE$<>"FORGET" THEN 950                 <147>
910 GOSUB 2630:FOR I=AN TO 1 STEP-1           <217>
920 IF BE$<>WO$(I) THEN NEXT I                <108>
930 IF I>AN THEN PRINT BE$" I CAN'T FIND":
    GOTO 700                                  <050>
935 :                                         <149>
940 VOC=AD(I):AN=I-1:GOTO 700                 <163>
950 IF BE$<>"(" THEN 980                      <117>
960 IF BE$<>")" THEN GOSUB 2630:GOTO 960      <027>
970 GOTO 700                                  <184>
975 :                                         <189>
```

Listing 1. Tiny-Forth-Compiler. Beachten Sie bitte die Eingabehinweise auf Seite 16.

```
980 IF BE$<>"EDIT" THEN 1020                    <027>
990 GOSUB 2630 :SC=VAL(BE$)                     <196>
1000 PRINT"SCREEN:";SC:GOSUB 3280               <163>
1010 IF BE$="-->"THEN ZE$="":SC=SC+1:GOTO
     1000                                       <005>
1012 GOTO 700                                   <226>
1015 :                                          <229>
1020 IF BE$<>"LOAD" THEN 1050                   <066>
1030 GOSUB 2630:SC=VAL(BE$)                     <238>
1040 BLOCK=1:Z1=0:GOSUB 3110:GOTO 700           <233>
1050 IF BE$<>"-->" THEN 1070                    <048>
1060 SC=SC+1:GOSUB 3110:COMP=1:BLOCK=1:Z1=
     0:GOTO 700                                 <225>
1070 :                                          <030>
1080 IF BE$<>"VARIABLE" THEN 1145               <137>
1085 GOSUB 2630:AN=AN+1:WO$(AN)=BE$             <243>
1090 AD(AN)=VOC:XX=VOC+8                        <237>
1095 GOSUB 3470:POKE VOC,169                    <027>
1100 POKE VOC+1,FN H(XX)                        <181>
1105 POKE VOC+2,162                             <127>
1110 POKE VOC+3,FN L(XX)                        <208>
1115 POKE VOC+4,32:POKE VOC+5,42                <253>
1120 POKE VOC+6,192:POKE VOC+7,96               <155>
1125 POKE VOC+8,FN L(X)                         <050>
1130 POKE VOC+9,FN H(X)                         <167>
1135 VOC=VOC+10                                 <237>
1140 GOTO 700                                   <100>
1145 :                                          <105>
1150 IF BE$<>"MEMORY" THEN 1220                 <167>
1155 GOSUB 2630:AN=AN+1:WO$(AN)=BE$             <057>
1160 AD(AN)=VOC                                 <061>
1165 GOSUB 3470:POKE VOC,169                    <097>
1170 POKE VOC+1,FN H(VOC+12)                    <205>
1175 POKE VOC+2,162                             <197>
1180 POKE VOC+3,FN L(VOC+12)                    <232>
1185 POKE VOC+4,32:POKE VOC+5,42                <067>
1190 POKE VOC+6,192:AD=VOC+12+XX                <244>
1195 POKE VOC+7,96                              <218>
1200 POKE VOC+8,FN L(AD):POKE VOC+9,FN H(A
     D)                                         <111>
1205 POKE VOC+10,FN L(XX):POKE VOC+11,FN H
     (XX)                                       <126>
1210 VOC=AD:GOTO 700                            <062>
1220 :                                          <180>
1230 IF BE$<>"CONSTANT" THEN 1280               <052>
1240 GOSUB 2630:A$=": "+BE$+" "                 <025>
1250 GOSUB 3470                                 <238>
1260 ZE$=A$+STR$(X)+" ;"+ZE$                    <110>
1270 GOTO 700                                   <230>
1280 :                                          <242>
1290 IF BE$<>"CLEAR" THEN 1350                  <152>
1300 GOSUB 2630:SC=VAL(BE$)                     <254>
1310 FOR ZE=0 TO 24                             <162>
1320 SC$(ZE)=""                                 <063>
1330 NEXT ZE:GOSUB 3220                         <005>
1340 GOTO 700                                   <046>
1350 :                                          <056>
1360 IF BE$="SAVE-SYSTEM"THEN 3510              <219>
1365 :                                          <071>
1370 IF BE$="LOAD-SYSTEM"THEN 3720              <159>
1380 :                                          <086>
1390 IF BE$<>"FLOPPY" THEN 1420                 <144>
1400 GOSUB 2630                                 <030>
1410 OPEN 1,8,15,BE$:CLOSE 1:GOTO 700           <181>
1420 :                                          <126>
1430 IF BE$<>"LIST" THEN 1520                   <088>
1440 GOSUB 2630:SC=VAL(BE$):GOSUB 3110          <018>
1450 INPUT"AUF DRUCKER (Y/N)";A$:A=3:IF A$
     ="Y"THEN A=4                               <097>
1460 OPEN 4,A,-7*(A=4)                          <220>
1470 FOR Z=0 TO 23                              <070>
1480 PRINT#4,RIGHT$(STR$(Z),2)":"SC$(Z)         <023>
1490 NEXT Z:CLOSE 4                             <124>
1500 IF A=3 THEN POKE 198,0:WAIT 198,1          <175>
1510 COMP=0:GOTO 700                            <198>
1520 :                                          <226>
1530 PRINT BE$" I CAN'T FIND":GOTO 700          <190>
1533 :                                          <239>
1535 REM ************************               <143>
1540 REM ***      COMPILER       ***            <216>
1545 REM ************************               <155>
1548 :                                          <000>
1550 GOSUB 2630:AN=AN+1:WO$(AN)=BE$             <200>
1560 AD(AN)=VOC:COMP=1                          <206>
1570 :                                          <022>
1580 GOSUB 2630                                 <212>
1590 FOR I=1 TO ANZ                             <174>
1600 IF BE$<>WO$(I) THEN NEXT I                 <026>
1610 AD=AD(I)                                   <144>
1615 :                                          <067>
1620 IF BE$<>"BEGIN" THEN 1640                  <215>
1630 ST(SP)=VOC:SP=SP+1:GOTO 1570               <147>
1635 :                                          <087>
1640 IF BE$<>"UNTIL" THEN 1730                  <091>
1650 POKE VOC,32                                <192>
1660 POKE VOC+1,180:POKE VOC+2,194              <254>
1670 POKE VOC+3,176:POKE VOC+4,3                <041>
1680 POKE VOC+5,76                              <190>
1690 SP=SP-1:AD=ST(SP):IF SP<0 THEN 65535       <250>
1700 POKE VOC+6,FN L(AD)                        <176>
1710 POKE VOC+7,FN H(AD)                        <042>
1720 VOC=VOC+8:GOTO 1570                        <124>
1725 :                                          <177>
1730 IF BE$=";" THEN POKE VOC,96:VOC=VOC+1
     :COMP=0:GOTO 700                           <150>
1735 :                                          <187>
1740 GOSUB 3030                                 <028>
1750 IF OK=0 THEN 1800                          <085>
1760 POKE VOC,169:POKE VOC+1,FN H(XX)           <082>
1770 POKE VOC+2,162:POKE VOC+3,FN L(XX)         <117>
1780 POKE VOC+4,32:POKE VOC+5,42                <154>
1790 POKE VOC+6,192:VOC=VOC+7:GOTO 1570         <174>
1800 :                                          <254>
1810 IF BE$<>"IF" THEN 1870                     <083>
1820 POKE VOC,32:POKE VOC+1,180                 <125>
1830 POKE VOC+2,194:POKE VOC+3,176              <146>
1840 POKE VOC+4,3:POKE VOC+5,76                 <049>
1850 ST(SP)=VOC+6:SP=SP+1                       <123>
1860 VOC=VOC+8:GOTO 1570                        <010>
1870 :                                          <068>
1880 IF BE$<>"ENDIF" THEN 1930                  <240>
1890 SP=SP-1:AD=ST(SP)                          <222>
1900 POKE AD,FN L(VOC)                          <247>
1910 POKE AD+1,FN H(VOC)                        <020>
1920 GOTO 1570                                  <130>
1930 :                                          <128>
1940 IF BE$<>"ELSE" THEN 2010                   <225>
1950 AD=ST(SP-1)                                <077>
1960 ST(SP-1)=VOC+1                             <023>
1970 POKE VOC,76:VOC=VOC+3                      <121>
1980 POKE AD,FN L(VOC)                          <071>
1990 POKE AD+1,FN H(VOC)                        <100>
2000 GOTO 1570                                  <210>
2010 :                                          <208>
2020 IF BE$="WHILE" THEN 1820                   <170>
2030 :                                          <228>
2040 IF BE$<>"REPEAT" THEN 2110                 <060>
2050 AD=ST(SP-1):A2=ST(SP-2)                    <184>
2060 SP=SP-1                                    <154>
2070 POKE VOC,76                                <110>
2080 POKE VOC+1,FN L(A2)                        <036>
2090 POKE VOC+2,FN H(A2)                        <230>
2100 VOC=VOC+3:GOTO 1980                        <106>
2110 :                                          <054>
2120 IF BE$<>"."+CHR$(34) THEN 2225             <144>
2125 A$="":ZE$=MID$(ZE$,2)                      <022>
2130 IF LEFT$(ZE$,1)<>CHR$(34) THEN A$=A$+
     LEFT$(ZE$,1):ZE$=MID$(ZE$,2):GOTO 213
     0                                          <185>
2135 ZE$=MID$(ZE$,2):A$=A$+CHR$(0)              <063>
2140 AD=VOC+10                                  <200>
2145 POKE VOC,169                               <040>
2150 POKE VOC+1,FN H(AD)                        <227>
2155 POKE VOC+2,162                             <161>
2160 POKE VOC+3,FN L(AD)                        <254>
2165 POKE VOC+4,32:POKE VOC+5,234               <165>
2170 POKE VOC+6,194:POKE VOC+7,76               <173>
2175 AD=VOC+10+LEN(A$)                          <012>
2180 POKE VOC+8,FN L(AD)                        <149>
2185 POKE VOC+9,FN H(AD)                        <010>
2190 VOC=VOC+10                                 <020>
2200 FOR I=0 TO LEN(A$)-1                       <150>
2205 POKE VOC+I,ASC(MID$(A$,I+1,1))             <036>
2210 IF LEFT$(ZE$,1)=" " THEN ZE$=MID$(ZE$
     ,2):GOTO 2210                              <173>
2215 NEXT I                                     <011>
2220 VOC=AD:GOTO 1570                           <190>
```

Listing 1. Tiny-Forth-Compiler (Fortsetzung)

```
2225 :                                          <169>
2230 IF BE$<>"TEXT"+CHR$(34) THEN 2320          <213>
2235 A$="":ZE$=MID$(ZE$,2)                      <132>
2240 IF LEFT$(ZE$,1)<>CHR$(34) THEN A$=A$+
     LEFT$(ZE$,1):ZE$=MID$(ZE$,2):GOTO 224
     0                                          <045>
2245 ZE$=MID$(ZE$,2):A$=A$+CHR$(0)              <173>
2250 AD=VOC+10                                  <054>
2255 POKE VOC,169                               <150>
2260 POKE VOC+1,FN H(AD)                        <081>
2265 POKE VOC+2,162                             <015>
2270 POKE VOC+3,FN L(AD)                        <108>
2273 POKE VOC+4,32:POKE VOC+5,42:POKE VOC+
     6,192                                      <097>
2275 POKE VOC+7,76                              <022>
2280 AD=VOC+10+LEN(A$)                          <117>
2285 POKE VOC+8,FN L(AD)                        <254>
2290 POKE VOC+9,FN H(AD)                        <115>
2295 VOC=VOC+10                                 <125>
2300 FOR I=0 TO LEN(A$)-1                       <250>
2305 POKE VOC+I,ASC(MID$(A$,I+1,1)):NEXT        <041>
2310 IF LEFT$(ZE$,1)=" " THEN ZE$=MID$(ZE$
     ,2):GOTO 2310                              <083>
2315 VOC=AD:GOTO 1570                           <031>
2320 :                                          <010>
2330 IF BE$<>"DO" THEN 2390                     <088>
2340 POKE VOC,32                                <120>
2350 POKE VOC+1,FN L(AD)                        <189>
2360 POKE VOC+2,FN H(AD)                        <056>
2370 VOC=VOC+3:ST(SP)=VOC                       <014>
2380 SP=SP+1:GOTO 1570                          <196>
2390 :                                          <080>
2400 IF BE$<>"LOOP" AND BE$<>"+LOOP" THEN
     2500                                       <222>
2410 POKE VOC,32                                <190>
2420 POKE VOC+1,FN L(AD)                        <003>
2430 POKE VOC+2,FN H(AD)                        <126>
2440 POKE VOC+3,176:POKE VOC+4,3                <049>
2450 SP=SP-1:AD=ST(SP)                          <018>
2460 POKE VOC+5,76                              <208>
2470 POKE VOC+6,AD-256*INT(AD/256)              <142>
2480 POKE VOC+7,INT(AD/256)                     <091>
2490 VOC=VOC+8:GOTO 1570                        <132>
2500 :                                          <190>
2510 IF BE$<>"(" THEN 2540                      <225>
2520 GOSUB 2630:IF BE$<>")" THEN 2520           <128>
2530 GOTO 1570                                  <232>
2540 :                                          <230>
2550 IF BE$=";S" THEN POKE VOC,96:VOC=VOC+
     1:GOTO 1570                                <005>
2560 :                                          <252>
2570 IF I>AN THEN PRINT BE$" I CAN'T FIND"
     :COMP=0:GOTO 700                           <030>
2575 :                                          <011>
2580 POKE VOC,32                                <106>
2590 POKE VOC+1,AD-256*INT(AD/256)              <133>
2600 POKE VOC+2,INT(AD/256)                     <083>
2610 VOC=VOC+3:GOTO 1570                        <234>
2615 :                                          <051>
2620 REM ************************               <133>
2630 REM ** HOLE BEFEHL IN BE$ **               <160>
2635 REM ************************               <148>
2637 :                                          <073>
2640 IF ZE$="" THEN GOSUB 2750                  <086>
2650 IF LEFT$(ZE$,1)=" "THEN ZE$=MID$(ZE$,
     2):GOTO 2650                               <108>
2660 BE$="":FOR I=1 TO LEN(ZE$)                 <059>
2670 IF LEFT$(ZE$,1)=" " THEN 2710              <174>
2680 BE$=BE$+LEFT$(ZE$,1)                       <085>
2690 ZE$=MID$(ZE$,2)                            <164>
2700 NEXT I                                     <244>
2710 RETURN                                     <228>
2720 :                                          <156>
2730 REM *************************              <068>
2740 REM *** HOLE ZEILE IN ZE$ ***              <027>
2750 REM *************************              <088>
2755 :                                          <191>
2760 IF BLOCK=1 THEN 2880                       <023>
2770 IF COMP=0 THEN PRINT"{2SPACE}OK."          <039>
2780 SYS 42336                                  <123>
2790 ZE$=""                                     <190>
2800 FOR Z=512 TO 600                           <099>
2810 A=PEEK(Z)                                  <003>
2820 IF A=0 THEN 2850                           <231>
2830 ZE$=ZE$+CHR$(A)                            <153>
2840 NEXT Z                                     <010>
2850 IF LEFT$(ZE$,1)=" "THEN ZE$=MID$(ZE$,
     2):GOTO 2850                               <183>
2860 IF ZE$="" THEN 2770                        <066>
2870 RETURN                                     <134>
2880 ZE$=SC$(Z1):PRINT RIGHT$(STR$(Z1),2);
     ":";ZE$                                    <219>
2890 IF LEN(ZE$)<2 THEN ZE$="({2SPACE})"        <071>
2900 Z1=Z1+1                                    <164>
2910 IF Z1=24 THEN BLOCK=0                      <186>
2920 RETURN                                     <184>
2980 :                                          <162>
2990 REM *************************              <236>
3000 REM **   WANDELE ZAHL UM   **              <118>
3010 REM **        IN XX        **              <069>
3020 REM *************************              <010>
3030 :                                          <212>
3040 OK=1:X=1                                   <172>
3050 IF LEFT$(BE$,1)="-" AND VAL(BE$)<0 TH
     EN BE$=MID$(BE$,2):X=-1:GOTO 3080          <007>
3060 IF LEFT$(BE$,1)>="0" AND LEFT$(BE$,1)
     <="9" THEN 3080                            <135>
3070 OK=0:RETURN                                <177>
3080 XX=VAL(BE$)*X                              <108>
3090 IF XX<0 THEN XX=(256*256)+XX               <059>
3100 RETURN                                     <110>
3103 :                                          <031>
3105 REM ************************               <191>
3110 REM *****  LADE SCREEN  *****              <122>
3115 REM ************************               <201>
3118 :                                          <046>
3120 OPEN 1,8,15                                <240>
3130 OPEN 2,8,2,"SCR"+STR$(SC)+",S,R"           <184>
3140 INPUT#1,A                                  <225>
3150 IF A<>0 THEN CLOSE 2:CLOSE 1:FOR I=0
     TO 24:SC$(I)="":NEXT I:RETURN              <215>
3160 FOR ZE=0 TO 24:B$=""                       <021>
3170 POKE 251,2:SYS 830                         <249>
3180 FOR I=512 TO 600:X=PEEK(I):IF X THEN
     B$=B$+CHR$(X):NEXT I                       <142>
3190 SC$(ZE)=B$                                 <044>
3200 NEXT ZE                                    <200>
3210 CLOSE 2:CLOSE 1:RETURN                     <099>
3213 :                                          <141>
3215 REM *************************              <207>
3220 REM *****  SAVE  SCREEN  *****             <055>
3225 REM *************************              <217>
3228 :                                          <156>
3230 OPEN 1,8,2,"@:SCR"+STR$(SC)+",S,W"         <013>
3240 FOR ZE=0 TO 24                             <058>
3250 PRINT#1,SC$(ZE)                            <253>
3260 NEXT ZE                                    <004>
3270 CLOSE 1:ZE$="":PRINT"{CLR}";:RETURN        <068>
3273 :                                          <201>
3275 REM **********************                 <112>
3280 REM **** EDIT A SCREEN ****                <079>
3285 REM **********************                 <122>
3288 :                                          <216>
3290 GOSUB 3400                                 <022>
3300 PRINT"{HOME}";:COMP=1                      <188>
3310 GOSUB 2750                                 <242>
3315 IF LEFT$(ZE$,1)="N" THEN GOSUB 2630:G
     OSUB 2630:SC=VAL(BE$):GOSUB 3420:GOTO
      3300                                      <046>
3320 IF LEFT$(ZE$,1)="E" THEN ZE$="":COMP=
     0:GOTO 3220                                <152>
3321 IF LEFT$(ZE$,1)<>"I" THEN 3330             <187>
3322 GOSUB 2630:GOSUB 2630:Z=VAL(BE$):IF Z
     <0 OR Z>23 THEN GOSUB 3420:GOTO 3300       <242>
3323 GOSUB 2630:A=VAL(BE$):IF A<0 OR A>23
     THEN GOSUB 3420:GOTO 3300                  <033>
3324 FOR I=22-A TO Z STEP-1:SC$(I+A)=SC$(I
     ):SC$(I)="":NEXT                           <070>
3325 GOSUB 3420:GOTO 3300                       <213>
3330 IF LEFT$(ZE$,1)="S" THEN ZE$="":PRINT
     "{CLR}";:COMP=0:RETURN                     <147>
3331 IF LEFT$(ZE$,1)<>"D" THEN 3337             <246>
3332 GOSUB 2630:GOSUB 2630:Z=VAL(BE$):IF Z
     <0 OR Z>23 THEN GOSUB 3420:GOTO 3300       <254>
3333 GOSUB 2630:A=VAL(BE$):IF A<0 OR A>23
     THEN GOSUB 3420:GOTO 3300                  <045>
3334 FOR I=Z TO 22-A:SC$(I)=SC$(I+A):SC$(I
     +A)="":NEXT                                <131>
```

Listing 1. Tiny-Forth-Compiler (Fortsetzung)

```
3335 GOSUB 3420:GOTO 3300                        <225>
3337 IF LEFT$(ZE$,1)="L" THEN GOSUB 3420:G
     OTO 3300                                    <128>
3340 ZE=VAL(ZE$)                                 <004>
3350 ZE$=MID$(ZE$,3)                             <078>
3360 IF ZE>9 THEN ZE$=MID$(ZE$,2)                <032>
3370 SC$(ZE)=ZE$                                 <141>
3380 GOSUB 2630:IF BE$="-->" THEN GOTO 322
     0                                           <037>
3390 GOTO 3310                                   <124>
3393 :                                           <067>
3395 REM ************************                <227>
3400 REM *****   LIST SCREEN   *****             <098>
3405 REM ************************                <237>
3408 :                                           <082>
3410 GOSUB 3110                                  <128>
3420 PRINT"{CLR}";                               <056>
3430 FOR ZE=0 TO 23                              <248>
3440 PRINT RIGHT$(STR$(ZE),2);":";               <157>
3450 PRINT LEFT$(SC$(ZE),38)                     <018>
3460 NEXT ZE:RETURN                              <184>
3463 :                                           <137>
3465 REM **********************                  <048>
3470 REM ** HOLE WERT VOM TOS **                 <047>
3475 REM **********************                  <058>
3480 AD=52992+PEEK(2)                            <065>
3490 X=PEEK(AD-1)+256*PEEK(AD-2)                 <138>
3500 POKE 2,PEEK(2)-2:RETURN                     <243>
3503 :                                           <177>
3505 REM **********************                  <088>
3510 REM ***   SAVE-SYSTEM   ***                 <056>
3515 REM **********************                  <098>
3518 :                                           <192>
3520 GOSUB 2630                                  <118>
3530 OPEN 1,8,15,"S:"+BE$+".*":CLOSE 1           <206>
3540 OPEN 2,8,2,BE$+".VOC,P,W"                   <175>
3550 PRINT#2,AN:PRINT#2,VOC                      <063>
3560 FOR ZE=39 TO AN                             <075>
3570 PRINT#2,WO$(ZE)                             <202>
3580 PRINT#2,AD(ZE)                              <160>
3590 NEXT ZE                                     <082>
3600 CLOSE 2:BE$=BE$+".CODE"                     <016>
3610 POKE 187,FN L(720):POKE 188,FN H(720)       <013>
3620 FOR I=1 TO LEN(BE$)                         <095>
3630 POKE 719+I,ASC(MID$(BE$,I,1))               <115>
3640 NEXT I:POKE 183,LEN(BE$)                    <233>
3650 POKE 186,8:POKE 185,1                       <199>
3660 POKE 251,FN L(BE):POKE 252,FN H(BE)         <173>
3670 POKE 780,251                                <205>
3680 POKE 781,FN L(VOC)                          <225>
3690 POKE 782,FN H(VOC)                          <007>
3700 SYS 216+256*255                             <020>
3710 GOTO 700                                    <130>
3713 :                                           <133>
3715 REM ***************************             <012>
3720 REM ****     LOAD SYSTEM     ****           <056>
3725 REM ***************************             <022>
3728 :                                           <148>
3730 GOSUB 2630                                  <074>
3740 OPEN 2,8,2,BE$+".VOC,P,R"                   <217>
3750 INPUT#2,AN,VOC                              <249>
3760 FOR ZE=39 TO AN                             <021>
3770 INPUT#2,WO$(ZE)                             <068>
3780 INPUT#2,AD(ZE)                              <026>
3790 NEXT ZE:CLOSE 2                             <162>
3800 SYS 50139,BE$+".CODE",8                     <110>
3810 GOTO 700                                    <230>
3813 :                                           <233>
3815 REM ***************************             <112>
3820 REM ***        DATA         ***             <160>
3825 REM ***************************             <122>
3828 :                                           <248>
3830 DATA 166,251, 32,198,255,160,  0, 32,
     207,255,201, 13,240,  7,153,  0             <161>
3840 DATA   2,200, 76, 69,  3,169,  0,153,
       0,  2, 76,204,255                         <248>
3850 FOR I= 830 TO 858:READ A:POKE I,A:Z=Z
     +A:NEXT I                                   <157>
3860 IF Z<>3379 THEN PRINT"{RVSON}FEHLER I
     N DATA{RVOFF}":END                          <107>
3870 RETURN                                      <118>
```

© 64'er

Listing 1. Tiny-Forth-Compiler (Schluß)

Hypra-Save

Hypra-Save ist eine Ergänzung zu Hypra-Load. Es speichert Programme 3- bis 5mal schneller und kann mit Hypra-Load verwendet werden.

Ein großer Nachteil der Diskettenstation VC 1541 ist die durch den seriellen Bus und durch das DOS V2.6 bedingte geringe Geschwindigkeit. Inzwischen gibt es mehrere Programme, die das Laden von Diskette beschleunigen. Mit der hier vorgestellten Routine geht jetzt auch das Speichern von Programmen mit dem C 64 wesentlich rascher.

Hypra-Save ist 3- bis 5mal so schnell wie die Originalroutine. Es verträgt sich mit Hypra-Load und vielen anderen, auch professionellen, Programmen und Basic-Erweiterungen. Zur Bedienung von Hypra-Save sollten Sie folgendes beachten: Die Eingabe muß mit dem MSE erfolgen. Nach dem Laden startet man es wie gewohnt mit RUN. Danach sollte man NEW eingeben, wenn man ein eigenes Programm schreiben will.

Hypra-Save kann Files mit oder ohne Verify speichern.

Gibt man vor dem Filenamen als erstes Zeichen einen Stern ein, so wird nicht verifiziert. Feststellbar an bis zu 5mal schnelleren Speicherzeiten. Mit Verify ist Hypra-Save etwa 3mal schneller als die Original-SAVE-Routine. Wer einen »25, WRITE ERROR« bisher nur aus der Literatur kennt, der kann getrost ohne Verify arbeiten. Selbstverständlich kann man weiterhin Programme überschreiben. Dann ist der Klammeraffe mit anzugeben. So überschreibt der Befehl SAVE "*@:name", 8 ein File, ohne die auf Diskette geschriebenen Blöcke zu prüfen, also ohne Verify. Hat der Computer alle Daten gesendet, wird im Gegensatz zur Original-SAVE-Routine nicht gewartet, bis das Laufwerk die Datei geschlossen hat. Dies macht sich besonders beim Überschreiben von Programmen bemerkbar. Die Floppystation arbeitet noch, während der Computer sich längst zurückgemeldet hat. Man darf die Diskette selbstverständlich nicht vor dem Erlöschen der roten LED aus dem Laufwerk nehmen.

Beim Speichern von Programmen mit dem Klammeraffen kommt die 1541 häufig ins »Schleudern«, wie Sie vielleicht aus eigener Erfahrung wissen. So kann es passieren, daß einige Programme nicht mehr geladen werden können. Löschen Sie daher ein Programm erst mit dem SCRATCH-Befehl und speichern Sie erst dann die neue Version ab.

Anstelle von RUN/STOP dient bei Hypra-Save 64 die RESTORE-Taste zum Abbrechen. Allerdings wird die Programmdatei dann nicht geschlossen und erscheint im Directory mit einem Stern. Möchte man die nicht geschlossene Datei löschen, so geht das nicht mit dem SCRATCH-, sondern nur mit dem VALIDATE-Befehl. Übrigens blinkt die rote LED nach dem Drücken von RESTORE. Liest man dann den Fehlerkanal aus, so erhält man die Meldung »51, OVERFLOW IN RECORD«, gefolgt von Track und Sektor des letzten Blocks. Diese Fehlermeldung weist sonst auf einen Übertragungsfehler hin, der von zu vielen Geräten am seriellen Bus herrühren kann.

Sollte man eine merkwürdige Fehlernummer wie zum Beispiel 61 oder 71 erhalten, hilft meist nur ein Aus- und Einschalten der Floppy. Nach STOP/RESTORE oder nach einem Reset ist der SAVE-Vektor zurückgesetzt. Hypra-Save läßt sich dann mit SYS 365 wieder aktivieren.

Das Programm verschiebt sich nach dem Starten. Der eine Teil von Hypra-Save liegt ab \$0150 bis \$0177, der andere von \$D000 bis \$D5FF im Speicher. Wird die SAVE-Routine aufgerufen, so wird über den geänderten SAVE-Vektor in die Initialisierungs-Routine nach \$0150 gesprungen. Dort wird geprüft, ob die anzusprechende Gerätenummer größer als 3

ist. Wenn ja, blendet die Routine das RAM unter den I/O-Bausteinen ein. Dann kopiert es den Speicherbereich von $C000 bis $C5FF, der nach dem Speichern wieder zurückgeschrieben wird, nach $D600 bis $DBFF (RAM) und verschiebt die eigentlichen SAVE-Routinen von $D000 bis $D5FF (ROM) nach $C000 bis $C5FF. Denn die I/O-Bausteine können nicht vom darunterliegenden RAM angesprochen werden.

Durch diese Speicherbelegungstechnik wird kein normalerweise sichtbares RAM verändert und nur ein minimaler Bereich im Stack von der Hypra-Save-Routine belegt. Vom Programmteil bei $C000 wird das File zuerst wie gewohnt durch die normalen IEC-Busroutinen geöffnet. Dann testet Hypra-Save ob kein Fehler aufgetreten ist und die Daten gesendet werden können. Ist kein Fehler aufgetreten, so wird die SAVE-Routine für das Diskettenlaufwerk in das Floppy-RAM geschrieben und gestartet. Dieser Programmteil liegt von $0146 bis $01D9 im Floppy-RAM. Er enthält hauptsächlich die Transferroutine, mit der sofort nach dem Aufrufen das Diskettenstationsprogramm übertragen, in den Bereich von $0300 bis $04A0 geschrieben und dann gestartet wird.

Der wesentlich zur Geschwindigkeitserhöhung beitragende Datentransfer beruht auf der gleichzeitigen Übertragung von je zwei Bit und auf dem erheblich eingeschränkten Handshakebetrieb, mit dem beide Geräte ihre Bereitschaft zum Transfer signalisieren.

Die hohe SAVE-Geschwindigkeit wird aber auch vom Schreibalgorithmus erzielt, der die für das eventuell durchzuführende Verify benötigte Zeit erheblich verkürzt. Hierzu ein Beispiel, um die Arbeitsweise dieses Algorithmus zu verdeutlichen:

Wenn sich der Schreib-/Lesekopf in einer halben Diskettenumdrehung über dem Header des gewünschten Blocks befindet, die zu schreibenden Daten im Puffer stehen und codiert sind, so hat die Originalroutine den Pufferinhalt nach einer halben Umdrehung geschrieben und nach einer weiteren ganzen Umdrehung verifiziert. Diese Methode benötigt 1,5 Umdrehungen pro Block, also drei für zwei Blöcke.

Hypra-Save geht anders vor. Es verwendet zwei Puffer. Nach einer halben Umdrehung wird der erste, nach einer weiteren halben der zweite Block geschrieben. Es wird dann noch eine Umdrehung gebraucht, um beide Sektoren zu verifizieren. Man benötigt also nur zwei Rotationen für zwei Blöcke, der Zeitgewinn beträgt etwa 0,1 Sekunden pro Block. Sind sämtliche Bytes übertragen, so stellt der Computer wie erwähnt das RAM von $C000 bis $C5FF wieder her, springt in die Digitalisierungsroutine, schaltet die ursprüngliche Speicherbelegung ein und kehrt ins aufrufende Programm zurück. Das Laufwerk schließt währenddessen das File, wofür es mehr Zeit brauchen kann, wenn beim Überschreiben das alte Programm gelöscht werden muß.

(Martin Pfost/hm)

```
PROGRAMM : HYPRA-SAVE        0801 0D3F
--------------------------------------

0801 : 0C 08 C1 07 9E 20 32 30   77
0809 : 36 32 00 00 00 78 A5 01   B5
0811 : 48 A9 34 85 01 A0 28 B9   15
0819 : 51 08 99 4F 01 88 D0 F7   46
0821 : A2 06 84 AC A9 D0 85 AD   0F
0829 : A9 A0 85 14 A9 08 85 15   21
0831 : B1 14 91 AC C8 D0 F9 E6   AF
0839 : 15 E6 AD CA D0 F2 20 6D   86
0841 : 01 68 85 01 58 A2 DA BD   79
0849 : A0 07 20 D2 FF E8 D0 F7   4A
0851 : 60 A5 BA C9 04 B0 03 4C   D6
0859 : ED F5 78 A6 01 A9 34 85   6D
0861 : 01 8E 9F D4 20 4F D4 AE   59
0869 : 9F D4 86 01 58 60 A9 50   04
0871 : 8D 32 03 A9 01 8D 33 03   5D
0879 : 60 0D 0D 09 0E 55 4C 54   2A
0881 : 52 41 53 41 56 45 20 36   ED
0889 : 34 20 41 4B 54 49 56 0D   8A
0891 : 49 4E 49 54 20 3A 20 53   D9
0899 : 59 53 20 33 36 35 0D BA   C1
08A1 : 8E A0 D4 AD 19 03 8D A1   8D
08A9 : D4 AD 18 03 8D A2 D4 A9   4F
08B1 : 9C 8D 18 03 A9 C1 8D 19   8B
08B9 : 03 AD 9F D4 85 01 AD 00   2C
08C1 : DD 8D 1F 01 58 A9 61 85   B0
08C9 : B9 A2 2C A0 00 B1 BB C9   03
08D1 : 2A D0 0A A2 4C C6 B7 E6   E2
08D9 : BB D0 02 E6 BC 8E A5 C3   B8
08E1 : A9 C1 48 A9 75 48 A5 B7   52
08E9 : D0 03 4C 10 F7 20 D5 F3   10
08F1 : 20 8F F6 A5 BA 20 0C ED   04
08F9 : A5 B9 20 B9 ED A9 00 85   F1
0901 : 90 20 DD ED 20 DD ED 20   BF
0909 : FE ED A5 90 F0 02 18 60   B9
0911 : A9 1A A2 C2 85 AC 86 AD   FB
0919 : A9 46 A2 01 85 14 86 15   EB
0921 : A2 05 A5 BA 20 0C ED A9   74
0929 : 6F 20 B9 ED A0 FD B9 15   DF
0931 : C1 20 DD ED C8 D0 F7 A5   76
0939 : 14 20 DD ED A5 15 20 DD   D2
0941 : ED A9 1E 20 DD ED A0 00   5E
0949 : B1 AC 20 DD ED C8 C0 1E   78
0951 : 90 F6 20 FE ED A5 AC 18   33
0959 : 69 1E 85 AC 90 02 E6 AD   D8
0961 : A5 14 18 69 1E 85 14 90   C3
0969 : 02 E6 15 CA D0 B4 A5 BA   3C
0971 : 20 0C ED A9 6F 20 B9 ED   03
0979 : A0 FB B9 1A C1 20 DD ED   39
0981 : C8 D0 F7 20 FE ED 78 A2   3A
0989 : 00 8E 00 DD AD 11 D0 29   85
0991 : EF 8D 11 D0 BD AE C2 9D   3D
0999 : 00 C5 E8 D0 F7 AD 00 DD   79
09A1 : 30 FB 20 FE C1 A2 00 BD   63
09A9 : AE C3 9D 00 C5 E8 D0 F7   77
09B1 : 20 FE C1 20 8E FB A2 01   1A
09B9 : A5 AC 9D 01 C5 E8 A5 AD   D1
09C1 : 9D 01 C5 E8 20 D1 FC B0   53
09C9 : 38 A5 AD C9 C0 90 1D C9   11
09D1 : C6 B0 19 69 16 85 AD A9   FA
09D9 : 34 85 01 B1 AC 48 AD 9F   49
09E1 : D4 85 01 A5 AD E9 15 85   F6
09E9 : AD 68 B0 02 B1 AC 9D 01   30
09F1 : C5 E8 20 DB FC E0 FF 90   A6
09F9 : CB 20 D1 FC B0 03 A9 FF   B2
0A01 : 2C A9 00 48 8D 00 C5 8E   18
0A09 : 01 C5 20 FE C1 A2 01 68   DB
0A11 : D0 B2 18 A9 00 48 AD 11   91
0A19 : D0 09 10 8D 11 D0 AD 1F   B0
0A21 : 01 8D 00 DD 78 A9 34 85   55
0A29 : 01 AD A2 D4 8D 18 03 AD   45
0A31 : A1 D4 8D 19 03 68 AE A0   32
0A39 : D4 9A 60 38 AD 1F 01 09   63
0A41 : 10 8D 1F 01 B0 CD B9 00   60
0A49 : C5 85 95 A2 00 2C 00 DD   A8
0A51 : 50 FB 2C 00 DD 10 05 E8   EE
0A59 : D0 F8 F0 B6 A9 10 8D 00   0A
0A61 : DD AA 66 95 6A 66 95 6A   E5
0A69 : 4A 4A 8D 00 DD 8A 66 95   33
0A71 : 6A 66 95 6A 4A 4A 8D 00   EE
0A79 : DD 8A 66 95 6A 66 95 6A   ED
0A81 : 4A 4A 8D 00 DD 8A 66 95   4B
0A89 : 6A 66 95 6A 4A 4A 8D 00   06
0A91 : DD EA EA EA A9 00 8D 00   CC
0A99 : DD C8 D0 AA 60 A0 00 98   A0
0AA1 : 59 00 C5 C8 D0 FA 85 14   A8
0AA9 : 20 A7 C1 88 A5 14 4C AA   A0
0AB1 : C1 4D 2D 57 4D 2D 45 B7   12
0AB9 : 01 A0 00 84 11 A9 02 8D   1C
0AC1 : 00 18 A9 04 2C 00 18 F0   BD
0AC9 : FB A9 00 8D 00 18 A2 03   9C
0AD1 : CA D0 FD A2 0A AD 00 18   15
0AD9 : 4A 6A 4A 66 85 0A 0A 66   55
0AE1 : 85 AD 00 18 4A 6A 4A 66   2E
0AE9 : 85 0A 0A 66 85 AD 00 18   B9
0AF1 : 4A 6A 4A 66 85 0A 0A 66   6D
0AF9 : 85 AD 00 18 8E 00 18 4A   36
0B01 : 6A 4A 66 85 0A 0A 66 85   70
0B09 : A5 85 91 30 45 11 85 11   F0
0B11 : C8 D0 AA 88 60 85 31 20   34
0B19 : 46 01 B1 30 48 20 4A 01   03
0B21 : 68 91 30 A5 11 F0 ED 4C   FB
0B29 : 43 E8 78 A9 0A 8D 00 18   71
0B31 : A2 00 88 D0 FD CA D0 FA   7F
0B39 : A9 0A 85 69 A9 00 85 30   87
0B41 : A9 03 20 A2 01 A9 04 20   76
0B49 : A2 01 4C 3D 04 A5 00 A2   D9
0B51 : 01 86 00 29 02 F0 10 A6   F0
0B59 : 98 86 32 20 2E 04 20 AF   A8
0B61 : 03 20 03 04 4C 69 F9 A2   F3
0B69 : 08 20 35 03 A2 0A 20 35   94
0B71 : 03 A2 08 20 F5 03 A2 0A   E2
0B79 : 20 F5 03 A5 8C 30 E8 4C   90
0B81 : 69 F9 86 32 86 98 A5 8C   AC
0B89 : 10 61 B5 83 D0 5D BD 31   79
0B91 : 04 20 A2 01 C8 A6 32 A9   4C
0B99 : 80 95 83 A5 80 95 00 A5   79
0BA1 : 81 95 01 B1 30 F0 1C 20   9F
0BA9 : 21 F1 A6 82 F6 B5 D0 02   21
0BB1 : F6 BB A0 00 A5 80 91 30   B2
0BB9 : C8 A5 81 91 30 A5 80 C5   A4
0BC1 : 22 F0 02 84 8C A6 32 A9   87
0BC9 : 00 85 30 85 33 85 2E 85   6C
0BD1 : 36 85 0C 85 50 A9 BB 85   CA
0BD9 : 34 BD 31 04 85 2F 20 E9   DF
0BE1 : F5 85 3A BD 32 04 20 A3   EA
0BE9 : F7 A6 32 B5 83 F0 52 20   C0
0BF1 : 2E 04 AD 00 1C 29 10 D0   7A
0BF9 : 03 4C 81 F5 20 10 F5 A2   E1
0C01 : 09 50 FE B8 CA D0 FA A9   7B
0C09 : FF 8D 03 1C AD 0C 1C 29   11
0C11 : 1F 09 C0 8D 0C 1C A9 FF   DF
0C19 : A2 05 8D 01 1C B8 50 FE   88
0C21 : B8 CA D0 FA A0 BB B1 0C   98
0C29 : 50 FE B8 8D 01 1C C8 D0   8E
0C31 : F5 B1 30 50 FE B8 8D 01   03
0C39 : 1C C8 D0 F5 50 FE 4C 00   DA
0C41 : FE 60 86 32 2C 24 04 86   58
0C49 : 98 B5 83 F0 F4 20 2E 04   CC
0C51 : 20 0A F5 A0 BB B1 0C 50   22
0C59 : FE B8 4D 01 1C D0 19 C8   65
0C61 : D0 F3 B1 30 50 FE B8 4D   18
0C69 : 01 1C D0 0C C8 C0 FD D0   5A
0C71 : F1 A6 32 A9 00 95 83 60   F3
0C79 : 4C C5 F6 BD 31 04 85 31   C9
0C81 : BD 32 04 85 0D 60 05 01   F3
0C89 : 06 04 58 20 19 F1 A9 84   7C
0C91 : D5 A7 F0 05 95 A7 20 42   B2
0C99 : D0 A9 40 8D F9 02 A9 01   58
0CA1 : 85 83 20 07 D1 90 03 4C   17
0CA9 : F8 CF 20 3E DE F6 B5 A9   28
0CB1 : 00 85 8B 85 8D A9 80 85   3B
0CB9 : 8C A5 80 85 06 A9 E0 85   25
0CC1 : 00 A5 00 30 FC F0 24 C9   15
0CC9 : 01 F0 EA A5 18 85 06 A5   C3
0CD1 : 19 85 07 A2 00 A9 B0 20   13
0CD9 : 7D D5 20 99 D5 A9 E2 20   F3
0CE1 : 7D D5 20 99 D5 A5 8C D0   E2
0CE9 : CC F0 CE 4C 23 DB A0 C0   80
0CF1 : A2 D6 20 6A D4 A0 D0 A2   2F
0CF9 : C0 20 6A D4 20 00 C0 48   94
0D01 : A0 D6 A2 C0 20 6A D4 68   46
0D09 : 60 AD FA FF 48 AD FB FF   E0
0D11 : 48 A9 FB 8D FA FF A9 D3   DC
0D19 : 8D FB FF 84 AD 86 15 A0   D9
0D21 : 00 84 AC 84 14 A2 06 B1   F1
0D29 : AC 91 14 C8 D0 F9 E6 AD   90
0D31 : E6 15 CA D0 F2 68 8D FB   0F
0D39 : FF 68 8D FA FF 60 B2 4C   95
```

MSE-Listing zu Hypra-Save. Beachten Sie bitte die Eingabehinweise auf Seite 18.

Das Grab des Pharao

Wagen Sie den gefährlichen Weg durchs alte Ägypten.

Ziel des Grafikadventures ist es, die goldene Totenmaske des Pharaos zu finden, wozu eine Pyramide erforscht werden muß, in der zahlreiche Fallen gegen Grabräuber verborgen sind. Am Anfang des Spiels befindet sich der Abenteurer jedoch noch mitten in der Wüste und muß die Pyramide erst einmal finden.

Das Programm besteht aus 58 Bildern, die sich aus verschiedenen Komponenten (Durchgänge, Türen, Treppen und so weiter) zusammensetzen. Aufgrund der Vielzahl und Ähnlichkeit der Räume ergibt sich ein wahrer Irrgarten. Die Bilder werden, um den Effekt von hochauflösender Grafik zu simulieren, aus über 80 neudefinierten Zeichen und 33 Sprites zur Darstellung der Gegenstände aufgebaut. Die Befehlsauswertung der Eingaben folgt getrennt in zwei »INPUT« Statements: Das heißt nach jedem eingegebenen Wort muß »RETURN« gedrückt werden, wodurch sich jedoch kein Nachteil ergibt.

Das Programm versteht folgende 18 Verben: gehe, zerstöre, öffne, klopfe, frage, hacke, töte, krieche, werfe, schiebe, drehe, nimm, verliere, ziehe, list, save, stop.

Da großer Wert auf die Grafik gelegt wurde, wuchs die Länge des Programmes und leider auch die Unübersichtlichkeit beträchtlich an, wodurch manche Stellen möglicherweise etwas umständlich programmiert wurden. Doch ich bin sicher, daß dies der Spielfreude keinen Abbruch tun wird.

Auf der Suche nach der goldenen Totenmaske des Pharao müssen Sie hier die erste Aufgabe lösen. Eine Menge Programmzeilen sind auf dem Weg in die Wüste, in der das Grab des Pharao steht, einzutippen. Damit Sie nicht hier schon »in der Wüste der Fehlermeldungen« stehen, ist auch dieses Programm mit dem Checksummer einzugeben. Liegt diese Hürde hinter Ihnen, wird Ihnen ein freundlicher Beduine, sobald Sie Ihn gefunden haben, den richtigen Weg weisen.

(Wolfgang Rausch/rg)

Programmbeschreibung:

Zeilen	Inhalt
10 — 13	Maschinenprogramm zur Verschiebung des Zeichengenerators
30 — 150	Einlesen der Felder (Arrays)
164 — 410	Information zu den Bildern und Eingabe eines Verbs
500 — 1910	Befehlsauswertung
5000 — 5910	Kommentare und Eingabe eines Nomens
10000 — 10900	Einlesen der Charakter-DATA
12000 — 12167	Einlesen der Sprite-DATA
19000 — 20900	Zeichnen der Bilder (Hintergrund)
21000 — 21018	Zeichnen der Durchgänge
21030 — 21038	Zeichnen der Türen (Sprites)
21050 — 21500	Zeichnen der Gegenstände (hauptsächlich Sprites)
30000 — 30040	Gegenstände, die man trägt, auflisten
34000 — 35060	Routinen zum Einlesen und Abspeichern eines Files
39900 — 40044	DATAs für Verben, Nomen, Spielfeld und Gegenstände
45000 — 49065	Vorspann
50105 — 52030	Kommentare bei Niederlage des Spielers und Routinen für bewegliche Gegenstände

Programmbeschreibung »Grab des Pharao«

```
0 REM *** PHARAOS GRAVE V.2 ***                <020>
1 REM       BY W.RAUSCH                        <090>
2 REM       MOERIKEWEG 73                      <000>
3 REM       8504 STEIN                         <134>
5 A1$="{HOME,18DOWN}"                          <243>
6 A2$="{HOME,21DOWN}"                          <149>
7 A3$="{HOME,22DOWN}"                          <004>
8 A4$="{HOME,23DOWN}"                          <218>
9 PRINT"{CLR}":V=53248:SI=54272:POKE V+21,
  0:POKE 657,128                               <228>
10 FOR I=832 TO 865:READ A:POKE I,A:NEXT       <220>
11 DATA 120,169,51,133,1,169,0,133,95          <094>
12 DATA 133,90,133,88,169,208,133,96,169,2
   40,133,89,169,224,133,91,32,191,163         <019>
13 DATA 169,55,133,1,88,96                     <091>
15 SYS 832:POKE 850,160:SYS 832:POKE 56576
   ,PEEK(56576)AND 252:POKE 53272,8            <225>
16 POKE 648,192                                <169>
20 GOTO 10000                                  <240>
22 PRINT"{CLR}SCRATCH GRAB":OPEN 1,8,15,"S
   :GRAB":CLOSE 1                              <141>
24 SAVE"GRAB",8:VERIFY"GRAB",8:LIST 22-24      <213>
30 DIM PN(58,6),IT$(54),IN(46,3),CO$(18)       <108>
40 FOR I=1 TO 18:READ CO$(I):NEXT              <059>
50 FOR I=1 TO 54:READ IT$(I):NEXT              <196>
55 RETURN                                      <113>
60 FOR I=866 TO 895:READ A:POKE I,A:NEXT:I
   F KK=1 THEN RETURN                          <175>
65 FOR I=1 TO 58                               <221>
70 READ PN(I,1),PN(I,2),PN(I,3),PN(I,4),PN
   (I,5),PN(I,6)                               <231>
80 NEXT                                        <090>
90 FOR I=1 TO 46                               <116>
100 READ IN(I,1),IN(I,2),IN(I,3)               <063>
110 NEXT                                       <120>
130 RETURN                                     <188>
150 AP=1                                       <110>
155 POKE 53280,11:POKE 53281,0                 <051>
160 PRINT"{CLR}":POKE V+21,0:POKE V+17,PEE
    K(V+17)AND 239:GOSUB 19000                 <081>
164 POKE 214,15:SYS 58640:PRINT"{RED}▔▔▔▔▔
    ▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔{WH
    ITE}"                                      <089>
165 POKE V+17,PEEK(V+17)OR 16                  <096>
170 PRINT"{UP}MOEGLICHE RICHTUNGEN :";         <086>
174 FOR I=1 TO 6                               <147>
175 IF PN(AP,I)>0 THEN ON I GOSUB 179,180,
    181,182,184,185                            <245>
176 NEXT:GOTO 210                              <205>
179 PRINT" W";:RETURN                          <092>
180 PRINT" O";:RETURN                          <220>
181 PRINT" N";:RETURN                          <205>
182 PRINT" S";:RETURN                          <031>
184 PRINT" H";:RETURN                          <112>
185 PRINT" R":RETURN                           <110>
210 PRINT A1$"BESONDERHEITEN: ";:N=0           <131>
220 FOR I=1 TO 46                              <246>
222 IF N>0 AND IN(I,1)=AP AND IN(I,3)=1 TH
    EN PRINT"{16RIGHT}"IT$(I)                  <019>
230 IF IN(I,1)=AP AND N=0 AND IN(I,3)=1 TH
    EN PRINT IT$(I):N=1                        <142>
240 NEXT                                       <250>
242 IF AP=35 AND X8=1 THEN 50100               <124>
245 IF AP=16 AND IN(14,1)=-1 AND IN(15,1)>
    0 THEN 50800                               <233>
246 IF AP=53 AND IN(45,1)=-1 THEN 50720        <212>
247 IF AP=50 AND IN(46,1)<>-1 THEN 50207       <227>
250 PRINT A3$:INPUT"KOMMANDO ";B$              <084>
260 FOR I=1 TO 18                              <160>
270 IF LEFT$(B$,4)=LEFT$(CO$(I),4)THEN 300 <011>
280 NEXT                                       <036>
300 IF AP=41 AND IN(41,1)<>-1 OR AP=42 AND
     IN(41,1)<>-1 THEN GOSUB 52000             <038>
310 IF C=3 THEN 50208                          <169>
410 IF AP=24 THEN GOTO 50204                   <026>
500 IF I<>1 THEN 730                           <091>
510 GOSUB 5540                                 <196>
530 GOTO 550                                   <062>
540 GOTO 5530                                  <180>
550 IF AP=2 OR AP=3 THEN X1=X1+1               <252>
560 IF X1=2 THEN GOTO 50190                    <193>
570 IF AP=10 AND IN(6,1)=-1 THEN 5530          <159>
590 IF AP=19 AND A$="R"AND IN(9,1)<>19 THE
    N GOTO 50203                               <116>
```

```
610 IF AP=31 AND A$="S"THEN GOTO 50202             <039>
620 IF AP=48 AND A$="S"OR AP=44 AND A$="W"
    THEN 50206                                     <025>
630 IF AP=35 AND IN(14,1)<>AP THEN 50100           <121>
640 IF AP=37 AND A$="O"AND IN(30,1)=37 THE
    N AP=41:GOTO 5500                              <178>
645 IF AP=37 AND IN(30,1)<>37 AND A$="O"TH
    EN 50200                                       <236>
650 IF AP=50 AND A$="W"AND IN(31,1)<>0 THE
    N 5530                                         <205>
655 IF AP=38 AND A$="N"AND IN(28,1)=0 AND
    IN(46,1)<>-1 THEN 50207                        <166>
660 IF A$="W"AND PN(AP,1)<>0 THEN AP=PN(AP
    ,1):GOTO 5500                                  <078>
670 IF A$="O"AND PN(AP,2)<>0 THEN AP=PN(AP
    ,2):GOTO 5500                                  <214>
680 IF A$="N"AND PN(AP,3)<>0 THEN AP=PN(AP
    ,3):GOTO 5500                                  <225>
690 IF A$="S"AND PN(AP,4)<>0 THEN AP=PN(AP
    ,4):GOTO 5500                                  <239>
700 IF A$="H"AND PN(AP,5)<>0 THEN AP=PN(AP
    ,5):GOTO 5500                                  <245>
710 IF A$="R"AND PN(AP,6)<>0 THEN AP=PN(AP
    ,6):GOTO 5500                                  <134>
720 GOTO 5530                                      <104>
730 IF I<>2 THEN 830                               <132>
735 IF IN(1,1)<>-1 THEN 5530                       <066>
739 X2=X2+1:IF X2>5 THEN 5475                      <032>
740 GOSUB 5540                                     <172>
770 IF AP=7 AND IN(3,1)=0 AND IN(4,1)=AP A
    ND A$="EINGA"THEN IN(4,1)=0:GOTO 5410          <037>
780 IF AP=13 AND A$="TUER"THEN GOTO 50211          <177>
790 IF AP=13 AND A$="WAND"AND IN(11,1)=0 T
    HEN IN(11,1)=13:GOTO 5420          0           <055>
800 IF AP=21 AND IN(16,1)>0 AND A$=IT$(16)
    THEN IN(16,1)=0:GOTO 5500                      <114>
810 IF AP=54 AND IN(40,1)>0 AND A$="AUSGA"
    THEN GOTO 49000                                <080>
820 GOTO 5530                                      <206>
830 IF I<>3 THEN 940                               <045>
840 GOSUB 5540:GOSUB 5000                          <106>
850 IF X<>32 AND X<>40 AND X<>38 THEN GOTO
    5530                                           <128>
860 IF X<>32 THEN 920                              <010>
870 IF AP=15 THEN PN(15,3)=16:IN(13,1)=0:G
    OTO 5500                                       <165>
875 IF AP=13 THEN 50211                            <132>
880 IF AP=21 AND IN(16,1)=0 THEN PN(21,3)=
    22:IN(17,1)=0:GOTO 5500                        <225>
881 IF AP=25 AND IN(43,1)=25 THEN IN(43,1)
    =0:PN(25,4)=19:IN(42,1)=0:PN(19,3)=25:
    GOTO 5500                                      <165>
882 IF AP=31 AND X5=1 AND IN(23,1)=AP THEN
    IN(23,1)=0:PN(31,3)=32:GOTO 5500               <117>
890 IF AP=21 AND IN(16,1)>0 THEN 50213             <167>
895 IF X5=1 AND AP=31 THEN PN(AP,3)=32:IN(
    31,1)=0:GOTO 5500                              <080>
900 IF AP=45 THEN PN(45,3)=50:IN(28,1)=0:G
    OTO 5500                                       <159>
905 IF AP=42 THEN PN(42,3)=40:IN(26,1)=0:G
    OTO 5500                                       <187>
906 IF AP=47 AND PN(AP,3)=0 THEN PN(AP,3)=
    52:IN(36,1)=0:GOTO 5500                        <217>
910 IF AP=38 THEN PN(38,3)=50:IN(28,1)=0:G
    OTO 5500                                       <059>
920 IF AP=50 THEN PN(50,1)=44:IN(31,1)=0:G
    OTO 5500                                       <239>
925 IF AP=53 AND Y3=0 AND X=38 THEN Y3=1:G
    OTO 50700                                      <183>
926 IF AP=57 AND X=40 AND Y4=0 AND IN(44,1
    )=-1 THEN IN(41,1)=AP:Y4=1:GOTO 5458           <181>
930 GOTO 5530                                      <060>
940 IF I<>4 THEN 1130                              <091>
950 GOSUB 5540                                     <128>
955 IF AP<8 OR A$<>IT$(53)THEN GOTO 5530           <201>
960 PRINT A2$:INPUT"{GREEN}WELCHE: W O N S
    {3SPACE}";C$                                   <149>
970 IF AP=13 AND IN(11,1)=0 AND C$="O"THEN
    GOTO 5430                                      <089>
980 IF AP<>13 OR AP=13 AND C$<>"O"THEN 110
    0                                              <192>
1000 GOTO 5530                                     <130>
1100 PRINT A3$"{GREEN}NICHTS PASSIERT{7SPA
     CE,WHITE}":GOSUB 5800:GOSUB 5900:GOTO
     164                                           <139>
1130 IF I<>5 THEN 1180                             <111>
1140 GOSUB 5540                                    <064>
1150 GOSUB 5000                                    <122>
1160 IF AP=5 AND X=2 AND IN(X,1)>0 THEN IN
     (2,1)=0:IN(X,2)=0:GOTO 5400                   <155>
1170 GOTO 5530                                     <046>
1180 IF I<>6 THEN 1220                             <203>
1182 IF IN(1,1)<>-1 THEN 5530                      <005>
1183 X2=X2+1:IF X2>5 THEN 5475                     <224>
1185 GOSUB 5540:GOSUB 5000                         <197>
1200 IF AP=7 AND X<>4 AND IN(X,1)=AP AND I
     N(X,1)>0 THEN IN(X,1)=0:IN(4,3)=1:GOT
     O 5500                                        <135>
1210 GOTO 5530                                     <086>
1220 IF I<>7 THEN 1270                             <071>
1230 GOSUB 5540:GOSUB 5000                         <242>
1240 IF X<>15 AND X<>5 THEN 5530                   <162>
1250 IF AP=16 AND IN(X,1)=AP THEN IN(X,1)=
     0:GOTO 5500                                   <048>
1255 IF AP=35 AND X=5 THEN 50100                   <014>
1260 GOTO 5530                                     <136>
1270 IF I<>8 THEN 1320                             <168>
1280 GOSUB 5540                                    <206>
1285 IF AP=7 AND IN(4,1)=0 AND A$=LEFT$(IT
     $(11),5)THEN AP=8:GOTO 5500                   <163>
1290 IF AP=13 AND A$=LEFT$(IT$(51),5)AND I
     N(11,1)=AP THEN AP=14:GOTO 5500               <093>
1300 IF AP=14 AND A$=LEFT$(IT$(51),5)THEN
     AP=13:GOTO 5500                               <237>
1310 GOTO 5530                                     <188>
1320 IF I<>9 THEN 1360                             <044>
1330 GOSUB 5540:GOSUB 5000                         <088>
1335 IF X<>9 AND IN(X,1)<>-1 THEN 5530             <255>
1340 IF AP=51 AND PN(AP,5)=0 THEN PN(AP,5)
     =54:IN(X,1)=AP:GOTO 5440                      <242>
1345 IF AP=37 THEN IN(X,1)=0:GOTO 5470             <230>
1350 GOTO 5530                                     <228>
1360 IF I<>10 THEN 1410                            <012>
1370 GOSUB 5540                                    <040>
1375 IF AP=27 THEN 1395                            <020>
1376 GOSUB 5000                                    <094>
1390 IF AP=40 AND Y2=0 AND IN(X,1)=AP THEN
     Y2=1:IN(35,1)=AP:PN(40,3)=47:GOTO 54
     50                                            <069>
1395 IF AP=27 AND A$="GRABR"AND IN(46,3)=0
     THEN IN(46,3)=1:GOTO 5480                     <076>
1400 GOTO 5530                                     <022>
1410 IF I<>11 THEN 1460                            <230>
1420 GOSUB 5540:GOSUB 5000                         <178>
1430 IF AP=10 AND X=7 AND IN(6,1)=-1 THEN
     50206                                         <222>
1440 IF AP=52 AND X=37 AND IN(34,1)=AP THE
     N PN(52,3)=53:IN(34,1)=0:GOTO 1680            <150>
1450 GOTO 5530                                     <072>
1460 IF I<>12 THEN 1510                            <117>
1470 GOSUB 5540                                    <140>
1490 IF X4=1 THEN AP=54:GOTO 5500                  <179>
1500 GOTO 5530                                     <122>
1510 IF I<>13 THEN 1580                            <095>
1520 GOSUB 5540                                    <190>
1525 IF GE>=3 THEN 5460                            <203>
1530 GOSUB 5000                                    <248>
1535 IF AP=16 AND X=14 THEN IN(X,1)=-1:IN(
     15,3)=1:GOTO 5500                             <121>
1536 IF AP=10 AND X=6 THEN IN(7,3)=1               <028>
1540 IF IN(X,1)=AP AND IN(X,2)>0 THEN IN(X
     ,1)=-1:GOTO 5500                              <087>
1550 IF IN(X,2)=0 THEN 5530                        <172>
1560 IF IN(X,1)<>AP THEN PRINT A3$"{GREEN}
     NICHT VORHANDEN{WHITE}":GOSUB 5800:GO
     SUB 5900:GOTO 164                             <141>
1570 GOTO 5530                                     <194>
1580 IF I<>14 THEN 1640                            <012>
1590 GOSUB 5540                                    <006>
1600 GOSUB 5000                                    <064>
1610 IF IN(X,1)<>-1 THEN PRINT A3$"{GREEN}
     ICH HABE DAS NICHT{WHITE}":GOSUB 5800
     :GOSUB 5900:GOTO 164                          <071>
1612 IF AP=19 AND A$="SEIL"AND IN(9,1)=-1
     THEN GOSUB 5455:GOTO 1620                     <020>
1615 IF AP=31 AND A$="DIAMA"AND IN(X,1)=-1
     THEN X5=1:GOSUB 5465:IN(20,2)=0:IN(2
     0,3)=0:GOTO 1620                              <235>
1620 IN(X,1)=AP:GOTO 5500                          <135>
```

Listing »Grab des Pharao«.
Beachten Sie bitte die Eingabehinweise auf Seite 16.

```
1640 IF I<>15 THEN 1700                                    <172>
1650 GOSUB 5540:GOSUB 5000                                 <154>
1655 IF X<>7 THEN 5530                                     <112>
1660 IF AP=10 AND IN(6,1)=-1 AND X9=0 THEN
     PN(AP,1)=11:IN(8,1)=0:X9=1:GOTO 1680                  <222>
1670 GOTO 5530                                             <038>
1680 PRINT A3$"{GREEN}DIE "IT$(32)" OEFFNE
     T SICH{WHITE}":GOSUB 5800:GOTO 160                    <235>
1700 IF I<>16 THEN 1750                                    <145>
1710 GOTO 30000                                            <166>
1750 IF I<>17 THEN 1800                                    <031>
1755 GOTO 34000                                            <019>
1800 IF I<>18 THEN 1900                                    <216>
1810 POKE V+21,0:PRINT CHR$(14),"{CLR,2DOW
     N,3RIGHT}SIE WOLLEN AUFGEBEN (J/N) ?"                 <032>
1820 GET A$:IF A$="J"THEN SYS 64738                        <047>
1830 IF A$="N"THEN PRINT CHR$(142):GOSUB 5
     600:GOTO 160                                          <232>
1840 IF A$=""OR A$=" "THEN 1820                            <065>
1900 PRINT A4$"UNBEKANNTES WORT":GOSUB 580
     0                                                     <047>
1910 PRINT A3$"{17SPACE}":PRINT A4$"{24SPA
     CE}":GOTO 164                                         <230>
5000 FOR X=1 TO 46                                         <064>
5010 IF LEFT$(A$,5)=LEFT$(IT$(X),5)THEN RE
     TURN                                                  <227>
5020 NEXT                                                  <204>
5030 PRINT A4$;TAB(20)"{CYAN}UNBEKANNTES W
     ORT{WHITE}":FOR T=1 TO 1000:NEXT                      <218>
5035 GOSUB 5900                                            <085>
5040 X=0:GOTO 164                                          <174>
5400 PRINT A3$"{GREEN}GEHE IN RICHTUNG DES
     "IT$(50)"{WHITE}":GOSUB 5800                          <231>
5401 PN(5,2)=6:GOTO 160                                    <220>
5410 PRINT A3$"{GREEN}DURCHGANG IN DER MAU
     ER{WHITE}":IN(4,1)=0:X6=1:GOSUB 5800:
     GOTO 5500                                             <196>
5420 PRINT A3$"{GREEN}EIN "IT$(51)" WIRD S
     ICHTBAR{WHITE}":GOSUB 5800:GOTO 160                   <042>
5430 PRINT A3$"{GREEN}ES KLINGT HOHL{11SPA
     CE,WHITE}":GOSUB 5800:GOTO 160                        <146>
5440 PRINT A3$"{GREEN}"IT$(9)" HAT SICH OB
     EN VERFANGEN{WHITE}":X4=1:GOSUB 5800:
     GOTO 160                                              <129>
5450 PRINT A3$"{GREEN}EIN "IT$(11)" WIRD S
     ICHTBAR{WHITE}":GOSUB 5800:GOTO 160                   <068>
5455 PRINT A3$"{GREEN}"IT$(9)" HAENGT HINU
     NTER{WHITE}":GOSUB 5800:RETURN                        <241>
5458 PRINT A3$"{GREEN}ER ENTHAELT EIM "IT$
     (41)"{WHITE,INST}":GOSUB 5800:GOTO 16
     0                                                     <106>
5460 PRINT A3$"{GREEN}SIE KOENNEN NICHT SO
      VIEL TRAGEN{WHITE}":GOSUB 5800:GOTO
     160                                                   <053>
5465 PRINT A3$"{GREEN}"IT$(20)" IST DAS FE
     HLENDE "IT$(52)" DER "IT$(6)"{WHITE}"
     :GOSUB 5800:RETURN                                    <155>
5470 PRINT A3$"{GREEN}"IT$(9)" IST HINUNTE
     RGEFALLEN{WHITE}":GOSUB 5800:GOTO 160                 <231>
5475 PRINT A3$"{GREEN}"RIGHT$(IT$(1),5)" I
     ST JETZT STUMPF{WHITE}":GOSUB 5800:GO
     SUB 5900:GOTO 164                                     <126>
5480 PRINT A3$"{GREEN}ER TRUG EINE "IT$(46
     )" BEI SICH{WHITE}":GOSUB 5800:GOTO 1
     60                                                    <090>
5500 GOSUB 5900                                            <042>
5510 PRINT A4$"{CYAN}IN ORDNUNG{WHITE}":FO
     R T=1 TO 1000:NEXT:IF I=13 THEN GE=GE
     +1                                                    <032>
5520 IF I=14 THEN GE=GE-1                                  <077>
5525 GOSUB 5600:GOTO 160                                   <116>
5530 GOSUB 5900                                            <072>
5535 PRINT A4$"{CYAN}NICHT MOEGLICH{WHITE}
     ":FOR T=1 TO 1000:NEXT:PRINT A4$"{15S
     PACE}":GOTO 164                                       <191>
5540 PRINT A4$;TAB(25):INPUT A$:A$=LEFT$(A
     $,5):RETURN                                           <160>
5600 POKE V+21,0:POKE V+28,0:POKE V+23,0:P
     OKE V+29,0:RETURN                                     <160>
5800 FOR T=1 TO 3000:NEXT:RETURN                           <194>
5900 PRINT A3$"{39SPACE}"                                  <250>
5910 PRINT A4$"{38SPACE}":RETURN                           <107>
10000 AD=57344                                             <088>
10010 READ X:IF X=-1 THEN GOTO 45000                       <255>
10020 CG=AD+X*8:FOR I=0 TO 7:READ Q:POKE C
      G+I,Q:NEXT:GOTO 10010                                <113>
10110 DATA 28,252,243,207,63,255,255,255,2
      55                                                   <224>
10120 DATA 30,255,255,255,255,252,243,207,
      63                                                   <114>
10130 DATA 31,255,255,255,255,255,255,15,2
      40                                                   <218>
10140 DATA 33,255,255,255,255,15,240,255,2
      55                                                   <004>
10150 DATA 35,255,255,15,240,254,254,254,2
      54                                                   <030>
10160 DATA 36,0,31,227,252,254,254,254,254                 <001>
10170 DATA 37,255,255,255,255,63,207,243,2
      52                                                   <113>
10180 DATA 38,63,207,243,252,255,255,255,2
      55                                                   <139>
10190 DATA 39,0,248,199,63,127,127,127,127                 <028>
10200 DATA 40,255,255,240,15,255,255,255,2
      55                                                   <236>
10210 DATA 41,255,255,255,255,240,14,254,2
      54                                                   <118>
10220 DATA 42,255,255,255,255,255,255,240,
      15                                                   <032>
10230 DATA 43,254,253,251,247,239,223,191,
      127                                                  <150>
10240 DATA 44,254,254,254,254,254,254,254,
      254                                                  <238>
10250 DATA 45,127,127,127,127,127,127,127,
      127                                                  <096>
10260 DATA 46,255,255,255,255,255,255,255,
      255                                                  <102>
10270 DATA 47,0,255,255,255,255,255,255,25
      5                                                    <158>
10280 DATA 48,0,254,254,254,254,254,254,25
      4                                                    <232>
10290 DATA 49,0,127,127,127,127,127,127,12
      7                                                    <006>
10300 DATA 50,127,191,223,239,247,251,253,
      254                                                  <048>
10400 DATA 51,239,239,239,239,239,223,191,
      127                                                  <157>
10410 DATA 52,224,239,239,239,239,239,239,
      239                                                  <011>
10420 DATA 53,239,239,239,143,47,239,239,2
      39                                                   <215>
10430 DATA 54,0,239,239,239,239,239,239,23
      9                                                    <137>
10440 DATA 55,254,254,254,254,254,224,235,
      239                                                  <246>
10450 DATA 56,255,255,255,255,255,255,95,2
      41                                                   <134>
10460 DATA 57,254,254,254,254,0,254,254,25
      4                                                    <248>
10470 DATA 27,239,239,239,239,239,239,239,
      224                                                  <172>
10480 DATA 59,239,239,239,239,247,251,253,
      254                                                  <016>
10490 DATA 60,239,239,239,239,239,239,239,
      15                                                   <065>
10500 DATA 61,239,239,227,233,236,239,239,
      239                                                  <255>
10510 DATA 62,239,239,239,239,15,239,239,2
      39                                                   <197>
10520 DATA 29,15,239,239,239,239,239,239,2
      39                                                   <235>
10530 DATA 64,127,127,127,127,0,127,127,12
      7                                                    <030>
10540 DATA 65,255,255,255,255,255,15,239,2
      39                                                   <194>
10550 DATA 67,0,255,255,0,0,0,0,0                          <225>
10560 DATA 68,239,239,239,239,239,239,239,
      239                                                  <144>
10570 DATA 69,254,254,254,254,254,254,240,
      15                                                   <207>
10580 DATA 70,239,239,239,239,240,239,239,
      239                                                  <164>
10590 DATA 71,254,254,254,254,254,254,254,
      0                                                    <066>
10600 DATA 72,127,127,127,127,127,127,127,
      0                                                    <160>
10610 DATA 73,15,240,255,255,255,255,255,2
      55                                                   <254>
10620 DATA 74,240,15,255,255,255,255,255,2
      55                                                   <236>
10630 DATA 75,255,170,255,170,255,170,255,
      170                                                  <210>
10640 DATA 76,85,85,85,85,85,87,95,255                     <249>
10650 DATA 77,85,85,85,87,127,255,255,255                  <167>
10660 DATA 78,85,85,85,85,245,255,255,255                  <179>
10670 DATA 79,85,85,85,85,85,245,255,255                   <105>
10680 DATA 80,85,85,85,85,85,85,245,255                    <081>
```

```
10685 DATA 81,63,207,243,252,254,254,254,2
      54                                   <048>
10690 DATA 82,254,254,254,254,15,240,255,2
      55                                   <036>
10695 DATA 83,127,255,15,240,255,255,255,2
      55                                   <156>
10700 DATA 84,252,242,206,62,254,254,254,2
      54                                   <249>
10705 DATA 85,254,254,254,254,252,243,207,
      63                                   <021>
10710 DATA 86,255,255,255,255,252,242,206,
      62                                   <051>
10715 DATA 87,255,255,255,255,240,14,254,2
      54                                   <243>
10720 DATA 88,254,254,254,254,254,254,240,
      15                                   <101>
10725 DATA 89,254,254,240,15,255,255,255,2
      55                                   <156>
10730 DATA 90,126,190,222,238,246,250,252,
      254                                  <204>
10735 DATA 91,127,127,127,127,63,207,243,2
      52                                   <004>
10740 DATA 92,14,240,255,255,255,255,255,2
      55                                   <000>
10745 DATA 93,96,99,105,108,110,110,110,11
      0                                    <049>
10750 DATA 94,6,198,150,54,118,118,118,118 <028>
10755 DATA 95,110,110,110,110,110,110,110,
      110                                  <133>
10760 DATA 97,118,118,118,118,118,118,118,
      118                                  <202>
10765 DATA 98,110,110,110,110,108,105,99,1
      5                                    <031>
10770 DATA 99,118,118,118,118,54,150,198,2
      40                                   <100>
10775 DATA 100,0,0,0,0,0,15,63,127         <165>
10780 DATA 101,15,31,63,127,255,255,255,25
      5                                    <138>
10785 DATA 102,240,248,252,254,255,255,255
      ,255                                 <105>
10790 DATA 103,0,0,0,0,0,240,252,254       <153>
10795 DATA 104,168,168,168,168,168,168,0,0 <018>
10800 DATA 105,170,170,170,170,170,170,0,0 <055>
10805 DATA 106,168,168,168,168,168,168,168
      ,168                                 <153>
10810 DATA 107,162,162,162,162,162,162,162
      ,162                                 <138>
10815 DATA 108,0,0,168,168,168,168,168,168 <167>
10820 DATA 109,170,170,170,170,170,170,170
      ,170                                 <009>
10825 DATA 110,0,0,170,170,170,170,170,170 <192>
10830 DATA 111,0,0,129,195,255,255,255,255 <228>
10835 DATA 112,255,255,248,6,184,191,255,2
      55                                   <196>
10840 DATA 113,0,0,1,3,15,255,255,255      <188>
10845 DATA 114,0,0,128,192,224,255,255,255 <221>
10850 DATA 115,0,0,0,255,255,0,0,0         <089>
10855 DATA 116,193,182,182,128,193,227,193
      ,193                                 <017>
10860 DATA 117,129,129,129,129,255,255,255
      ,255                                 <222>
10865 DATA 118,255,219,129,195,195,129,219
      ,255                                 <055>
10900 DATA-1                               <114>
12000 FOR I=0 TO 62:READ Q:POKE 50816+I,Q:
      NEXT                                 <028>
12001 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
      ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
      ,0                                   <247>
12002 DATA 0.0.64.0.0,128,0,1,0,0,3,255,24
      8,4,0,0,8,0,0,8,0,0,8,0,0,0,0,0      <185>
12004 FOR I=0 TO 62:READ Q:POKE 50880+I,Q:
      NEXT                                 <022>
12005 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
      ,0,0,0,32,0,0,120,0,0,254,0,1,255,12
      8                                    <015>
12006 DATA 3,255,224,7,255,248,15,255,254,
      31,255,252,63,255,248,127,255,240,0,
      0                                    <128>
12007 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0       <202>
12010 FOR I=0 TO 62:READ Q:POKE 50944+I,Q:
      NEXT                                 <037>
12011 DATA 0,0,64,0,64,128,2,33,0,1,36,128
      ,0,200,76,17,8,80,18,69,32,34,38,64,
      36                                   <014>
12012 DATA 100,48,20,162,64,8,164,128,4,15
      3,192,5,22,40,2,20,16,5,38,0,8,169,0
      ,4                                   <047>
12013 DATA 72,192,0,80,0,0,96,0,0,96,0,0,9
      6,0                                  <232>
12015 FOR I=0 TO 62:READ Q:POKE 51008+I,Q:
      NEXT                                 <169>
12016 DATA 3,252,0,2,168,0,3,92,0,3,92,0,3
      ,92,0,3,92,0,15,255,0,15,255,0,15,25
      5                                    <217>
12017 DATA 0,15,255,208,15,255,208,15,252,
      0,14,168,0,7,252,0,7,252,0,3,252,0,3 <061>
12018 DATA 252,0,2,138,0,2,138,0,0,163,128
      ,0,0,0                               <228>
12020 FOR I=0 TO 62:READ Q:POKE 51072+I,Q:
      NEXT                                 <164>
12021 DATA 0,0,0,127,251,254,127,251,254,1
      27,249,254,0,0,0,127,63,60,127       <206>
12022 DATA 127,190,127,127,190,62,127,190,
      0,0,0,127,231,254,127,239,254,127,23
      9                                    <245>
12023 DATA 254,127,207,252,0,0,0,57,255,15
      8,125,255,222,125,255,222,125,255,22
      0                                    <059>
12024 DATA 0,0,0,255,255,255               <215>
12026 FOR I=0 TO 62:READ Q:POKE 51136+I,Q:
      NEXT                                 <179>
12027 DATA 95,255,250,95,255,250,80,0,10,8
      0,0,10,88,0,26,88,0,26,92,0,58,92,0,
      58                                   <238>
12028 DATA 94,0,122,94,0,122,95,0,250,95,0
      ,250,95,129,250,95,126,250,94,255,12
      2                                    <231>
12029 DATA 93,255,186,91,255,218,87,255,23
      4,79,255,242,127,255,254,0,0,0       <186>
12031 FOR I=0 TO 62:READ Q:POKE 51200+I,Q:
      NEXT                                 <155>
12032 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
      ,31,0,0,14,0,0,14,0,0,27,0,0,42,128,
      0                                    <081>
12033 DATA 113,192,0,106,192,0,59,128,0,21
      ,0,0,14,0,0,0,0,0,0,0,0,0,0,0,0,0,
      0                                    <217>
12034 DATA 0,0,0,0                         <102>
12036 FOR I=0 TO 62:READ Q:POKE 51264+I,Q:
      NEXT                                 <190>
12038 DATA 0,0,0,0,0,0,0,0,0,1,85,128,2,17
      0,64,2,170,0,2,175,0,2,173,0,2,174,1
      92                                   <202>
12039 DATA 2,175,192,2,175,0,3,253,0,3,255
      ,0,0,255,0,0,242,0,0,242,0,0,242,0,3 <010>
12040 DATA 252,0,3,252,0,0,0,0,0,0,0       <181>
12042 FOR I=0 TO 62:READ Q:POKE 51328+I,Q:
      NEXT                                 <205>
12043 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,8
      ,0,0,16,6,0,30,57,120,31,192,112,16,
      6                                    <033>
12044 DATA 12,8,121,194,3,128,57,28,120,5,
      33,135,130,24,112,28,7,14,98,0,225,1
      40                                   <092>
12045 DATA 0,30,48,0,1,192,0,0,0,0,0,0     <226>
12047 FOR I=0 TO 62:READ Q:POKE 51392+I,Q:
      NEXT                                 <200>
12048 DATA 0,0,0,15,255,240,10,0,80,10,0,8
      0,10,0,80,19,255,200,20,0,40,20,0,40 <240>
12049 DATA 20,0,40,39,255,228,40,0,20,40,0
      ,20,40,0,20,79,255,242,80,0,10,80,0,
      10                                   <037>
12050 DATA 80,0,10,159,255,249,160,0,5,160
      ,0,5,255,255,255                     <255>
12052 FOR I=0 TO 62:READ Q:POKE 51456+I,Q:
      NEXT                                 <214>
12053 DATA 0,0,0,0,0,0,0,0,0,0,0,0,56,0,
      0,76,0,0,134,0,0,18,64,0,47,160,0,11
      0                                    <073>
12054 DATA 160,0,191,64,0,156,64,0,56,0,0,
      68,0,1,132,0,2,8,0,5,20,0,0,0,0,0,0,
      0                                    <252>
12055 DATA 0,0,0,0,0,0                     <116>
12057 FOR I=0 TO 62:READ Q:POKE 51520+I,Q:
      NEXT                                 <190>
12058 DATA 16,0,0,21,0,0,31,64,0,31,80,0,3
      1,213,0,31,254,64,31,255,64,31,255,6
      4                                    <069>
12059 DATA 31,255,208,31,255,244,21,85,84,
      7,255,244,1,255,244,0,127,244,0,23,2
      44                                   <117>
12060 DATA 0,5,244,0,1,244,0,0,116,0,0,20,
      0,0,4,0,0,0                          <096>
12062 FOR I=0 TO 62:READ Q:POKE 51584+I.Q:
```

Listing »Grab des Pharao« (Fortsetzung)

```
      NEXT                                    <223>
12063 DATA 0,0,16,0,0,80,0,1,240,0,7,208,0
      ,7,208,0,95,208,1,255,208,5,255,208     <156>
12064 DATA 31,255,208,31,255,208,21,85,80,
      31,255,64,31,253,0,23,212,0,31,208,0    <061>
12065 DATA 31,64,0,29,0,0,20,0,0,16,0,0,0,
      0,0,0,0,0                               <052>
12067 FOR I=0 TO 62:READ Q:POKE 51648+I,Q:
      NEXT                                    <237>
12068 DATA 0,0,0,0,0,0,0,0,0,0,0,3,255,1
      92,2,0,64,2,239,64,2,239,64,2,15,64,
      2                                       <202>
12069 DATA 240,64,2,247,64,2,247,64,2,7,64
      ,2,224,64,2,239,64,2,239,64,2,0,64,2    <135>
12070 DATA 247,64,2,247,64,2,0,64,3,255,19
      2                                       <208>
12072 FOR I=0 TO 62:READ Q:POKE 51712+I,Q:
      NEXT                                    <213>
12073 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
      ,0,0,0,0,0,0,0,0,0,8,0,0,55,0,0,59,0
      ,0                                      <225>
12074 DATA 213,192,0,191,128,0,204,192,0,1
      91,128,0,255,192,0,183,128,0,255,192
      ,0                                      <215>
12075 DATA 55,0,0,55,0,0,55,0                 <248>
12077 FOR I=0 TO 62:READ Q:POKE 51776+I,Q:
      NEXT                                    <246>
12078 DATA 0,0,0,0,56,0,0,124,0,0,84,0,0,1
      24,0,0,40,0,0,56,0,1,255,0,2,214,128
      ,2                                      <235>
12079 DATA 238,128,2,214,128,3,57,128,0,25
      4,0,0,254,0,0,124,0,0,108,0,0,108,0,
      0                                       <140>
12080 DATA 108,0,0,108,0,0,108,0,0,230,0      <095>
12082 FOR I=0 TO 62:READ Q:POKE 51840+I,Q:
      NEXT                                    <222>
12083 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
      ,0,0,0,0,0,0,0,0,0,0,0,28,0,0,62,0
      ,0                                      <206>
12084 DATA 28,0,0,0,0,0,0,0,0,0,0,0,0,0,
      0,0,0,0,0,0,0,0,0,0,0,0,0               <249>
12086 FOR I=0 TO 62:READ Q:POKE 51904+I,Q:
      NEXT                                    <235>
12087 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,2
      25,128,0,30,3,128,0,48,112,1,204,15,
      14                                      <092>
12088 DATA 6,0,192,0,0,51,193,192,12,14,56
      ,0,48,6,0,0,0,0,0,0,0,0,0,0,0,0,
      0                                       <021>
12089 DATA 0,0,0,0,0,0                        <150>
12091 FOR I=0 TO 62:READ Q:POKE 51968+I,Q:
      NEXT                                    <012>
12092 DATA 0,0,0,0,0,0,0,0,0,0,0,0,21,84
      ,0,74,148,1,42,100,4,169,164,18,166     <078>
12093 DATA 164,85,90,164,106,154,164,106,1
      54,144,106,154,124,106,153,205,106,1
      51                                      <002>
12094 DATA 240,85,80,60,0,0,13,0,0,0,0,0,0
      ,0,0,0,0,0,0                            <093>
12096 FOR I=0 TO 62:READ Q:POKE 52032+I,Q:
      NEXT                                    <107>
12097 DATA 255,255,255,128,36,1,128,60,1,1
      28,66,1,128,126,1,128,129,1,128,255,
      1                                       <167>
12098 DATA 129,0,129,129,255,129,130,0,65,
      131,255,193,132,0,33,135,255,225,136    <200>
12099 DATA 0,17,143,255,241,144,0,9,159,25
      5,249,160,0,5,191,255,253,192,0,3,25
      5                                       <061>
12100 DATA 255,255                            <253>
12102 FOR I=0 TO 62:READ Q:POKE 52096+I,Q:
      NEXT                                    <141>
12103 DATA 0,0,0,0,28,0,0,3,128,0,0,64,0,1
      ,128,0,14,0,0,48,0,0,64,0,0,48,0,0,1
      2                                       <204>
12104 DATA 0,0,3,0,0,0,128,0,1,0,0,6,0,0,2
      4,0,0,96,0,0,224,0,0,192,0,0,0,0,0,0    <032>
12105 DATA 0,0,0,0                            <173>
12107 FOR I=0 TO 62:READ Q:POKE 52160+I,Q:
      NEXT                                    <117>
12108 DATA 0,6,0,0,24,0,0,32,0,0,48,0,0,12
      ,0,0,6,0,0,1,0,0,3,0,0,12,0,0,48,0,0
      ,64                                     <126>
12109 DATA 0,0,96,0,0,24,0,0,6,0,0,24,0,0,
      96,0,0,224,0,0,192,0,0,0,0,0,0,0,0,0
      ,0                                      <207>
12111 FOR I=0 TO 62:READ Q:POKE 52224+I,Q:
      NEXT                                    <130>
12112 DATA 0,168,42,160,170,10,0,42,2,160,
      47,0,0,15,0,160,12,0,0,40,0,0,255,0,
      3                                       <072>
12113 DATA 60,192,163,60,240,3,252,0,160,6
      0,0,0,42,240,160,42,252,0,85,12,0,85    <230>
12114 DATA 12,0,65,12,160,65,12,0,65,15,16
      0,65,5,0,0,0                            <209>
12116 FOR I=0 TO 62:READ Q:POKE 52288+I,Q:
      NEXT                                    <163>
12117 DATA 160,0,0,128,42,0,0,170,10,0,168
      ,0,0,240,10,0,240,0,0,48,10,0,235,0,
      3                                       <131>
12118 DATA 60,192,3,60,202,15,60,192,0,60,
      202,0,60,0,3,168,0,15,168,10,48,85,0    <174>
12119 DATA 48,85,10,48,65,0,112,65,10,80,6
      5,0,0,0,0                               <002>
12121 FOR I=0 TO 62:READ Q:POKE 52352+I,Q:
      NEXT                                    <139>
12122 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
      ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0      <079>
12123 DATA 0,0,63,255,252,127,255,254,255,
      255,255,255,255,255,0,0,0,0,0,0,0,
      0.0                                     <223>
12124 DATA 0,0,0,0,0                          <241>
12125 FOR I=0 TO 62:READ Q:POKE 52416+I,Q:
      NEXT                                    <152>
12126 DATA 0,0,0,3,192,0,7,192,0,7,128,0,7
      ,128,0,2,128,0,7,0,0,59,128,0,47,192
      ,0                                      <065>
12127 DATA 7,224,0,14,0,0,13,240,0,14,248,
      0,7,124,0,7,190,0,5,111,0,2,139,128,
      2                                       <215>
12128 DATA 130,192,7,129,64,31,255,252,31,
      255,252                                 <212>
12130 FOR I=0 TO 62:READ Q:POKE 52480+I,Q:
      NEXT                                    <147>
12131 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
      ,0,0,0,0,0,0,0,0,2,170,128,10,170,16
      0                                       <227>
12132 DATA 9,221,224,11,119,96,10,170,160,
      11,170,96,9,221,224,11,119,96,10,170    <126>
12133 DATA 160,8,0,32,8,0,32,8,0,32,0,0,0     <241>
12139 FOR I=0 TO 62:READ Q:POKE 52544+I,Q:
      NEXT                                    <165>
12140 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
      ,0,0,0,0,0,0,0,0,0,0,0,8,0,1,214,0
      ,2                                      <125>
12141 DATA 33,0,4,76,128,5,144,128,4,139,0
      ,4,100,0,7,24,0,3,128,0,1,128,0,0,0,
      0                                       <021>
12142 DATA 0,0,0                              <070>
12144 FOR I=0 TO 62:READ Q:POKE 52608+I,Q:
      NEXT                                    <179>
12145 DATA 0,0,0,0,0,0,120,0,0,48,0,0,48
      ,0,0,120,0,0,180,0,1,50,0,2,121,0,4     <153>
12146 DATA 120,128,8,252,64,8,212,64,9,182
      ,64,5,122,128,3,119,0,3,183,0,3,203,
      0                                       <047>
12147 DATA 1,222,0,0,220,0,0,120,0,0,0,0      <111>
12149 FOR I=0 TO 62:READ Q:POKE 52672+I,Q:
      NEXT                                    <174>
12150 DATA 0,0,0,0,0,0,255,192,0,255,192
      ,0,255,192,1,128,96,1,128,96,1,128,9
      6                                       <202>
12151 DATA 3,0,48,3,0,48,3,0,48,6,0,24,6,0
      ,24,6,0,24,12,0,12,12,0,12,12,0,12,1
      5                                       <023>
12152 DATA 255,252,13,154,236,10,205,148,1
      5,255,252                               <152>
12154 FOR I=0 TO 62:READ Q:POKE 52736+I,Q:
      NEXT                                    <188>
12155 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
      ,42,0,0,21,0,0,38,0,0,21,0,0,42,0,0,
      21                                      <060>
12156 DATA 0,0,170,128,0,85,64,0,170,128,0
      ,85,64,0,170,128,0,85,64,0,0,0,0,0,0
      ,0                                      <007>
12157 DATA 0,0,0,0,0                          <018>
12159 FOR I=0 TO 62:READ Q:POKE 52800+I,Q:
      NEXT                                    <164>
12160 DATA 0,0,0,0,0,0,127,128,0,125,128
      ,0,118,128,0,239,192,0,237,192,0,254    <218>
12161 DATA 192,1,166,224,1,223,224,1,254,2
      24,3,157,112,3,127,240,3,245,240,6,2
      51                                      <016>
12162 DATA 184,7,55,24,7,255,248,0,0,0,0,0
      ,0,0,0,0,0,0,0                          <118>
12164 FOR I=0 TO 62:READ Q:POKE 52864+I,Q:
```




```
       NEXT                                       <197>
12165 DATA 0,24,0,0,90,0,0,126,0,0,126,0,0
      ,126,0,0,60,0,0,60,0,0,60,0,0,24,0,1 <238>
12166 DATA 255,128,3,110,192,3,118,192,3,1
      10,192,1,223,128,0,126,0,0,126,0,0,1
      02                                   <016>
12167 DATA 0,0,102,0,0,102,0,1,255,128,7,2
      55,224                               <067>
15000 RETURN                               <070>
19000 IF AP>7 THEN 19100                   <202>
19001 IF AP=7 THEN 19200                   <239>
19002 IF AP=6 THEN GOSUB 19035             <238>
19003 POKE 53270,PEEK(53270)OR 16:POKE 532
      83,9:POKE 53282,6                    <190>
19004 PRINT TAB(10)"{BLUE}................
      ...."                                <137>
19006 PRINT TAB(10)"...................."  <150>
19008 PRINT TAB(10)"...................."  <152>
19010 PRINT TAB(10)".....{GREY 3}LMNOP{LIG
      .BLUE}...{GREY 3}LOP{LIG.BLUE}...."  <094>
19012 FOR T=1 TO 10:PRINT TAB(10)"{GREY 3}
      KKKKKKKKKKKKKKKKKKKK":NEXT           <177>
19031 GOSUB 21050:RETURN                   <135>
19035 POKE V+21,PEEK(V+21)OR 128:POKE 5017
      5,27:POKE V+14,145:POKE V+15,90:POKE
       V+46,9                              <108>
19036 POKE V+23,PEEK(V+23)OR 128:POKE V+29
      ,PEEK(V+29)OR 128:RETURN             <020>
19100 IF AP=15 OR AP=18 OR AP=19 OR AP=20
      OR AP=21 OR AP=29 OR AP=46 OR AP=52
      THEN 19110                           <039>
19102 GOTO 19999                           <133>
19110 PRINT TAB(12)"{BROWN}&%............↑
      £"                                   <084>
19111 PRINT TAB(12)"..@%........V£.."      <036>
19112 PRINT TAB(12)"..,.@%....V£,..."      <147>
19113 PRINT TAB(12)"I#R←,.1//0,.XH(J"      <250>
19114 PRINT TAB(12)".,.,=S-CC↑YJ-,.."      <077>
19115 PRINT TAB(12)".,.,..¥{2SPACE}K..-,..
      "                                    <081>
19116 PRINT TAB(12)"//0/0/¥{2SPACE}K/1/1//
      "                                    <033>
19117 PRINT TAB(12)"..,.U£TG+T&±.-.."      <051>
19118 PRINT TAB(12)"..UT.+....2,&±.."      <061>
19120 PRINT TAB(12)"↑I.,+......Z.,&%"      <112>
19121 PRINT TAB(12)".,.+.........2,.."     <123>
19122 PRINT TAB(12)".,+...........Z.."     <029>
19123 PRINT TAB(12)".+.............2."     <147>
19124 PRINT TAB(12)"+..............2"      <152>
19129 IF AP=19 THEN 20015                  <205>
19130 GOSUB 21032:RETURN                   <106>
19200 POKE 53270,PEEK(53270)OR 16:POKE 532
      82,7:POKE 53283,9                    <202>
19204 PRINT TAB(10)"{BROWN}ZZZZNZZZZNZZZZN
      ZZZZZ"                               <203>
19205 PRINT TAB(10)"ZZZZNZZZZNZZZZNZZZZZ"  <222>
19206 PRINT TAB(10)"££££?££££?££££?£££££"  <074>
19207 FOR T=1 TO 3:PRINT TAB(10)"ZZQZZZZQZ
      ZZZQZZZZQZZ":NEXT                    <131>
19210 PRINT TAB(10)"SSSSDSSS{3SPACE}SSSDSS
      SSS"                                 <079>
19211 PRINT TAB(10)"ZZZZNZZZ{3SPACE}ZZZNZZ
      ZZZ"                                 <055>
19212 PRINT TAB(10)"££££?£££{3SPACE}£££?££
      £££"                                 <102>
19213 FOR T=1 TO 4:PRINT TAB(10)"{GREY 3}K
      KKKKKKKKKKKKKKKKKKK":NEXT            <206>
19217 IF X6=1 THEN POKE V+21,PEEK(V+21)OR
      128:POKE 50175,31:POKE V+14,168:POKE
       V+15,108                            <132>
19218 POKE V+46,9                          <196>
19219 GOSUB 21050                          <238>
19220 RETURN                               <228>
19999 POKE 53270,PEEK(53270)AND 239        <244>
20000 PRINT TAB(10)"{BROWN}&%.............
      ...↑£"                               <247>
20001 PRINT TAB(10)"..&%............↑£.."  <165>
20002 PRINT TAB(10)"....1///1///1//0...."  <054>
20003 PRINT TAB(10)"I#!←...-...-..,*)(J"   <250>
20004 PRINT TAB(10)".,..1/1///1///10.,.."  <023>
20005 PRINT TAB(10)".,..-.-...-...-,.,.."  <232>
20006 PRINT TAB(10)"////1///1///1//0////"  <088>
20007 PRINT TAB(10)"....-...-...-..,...."  <101>
20008 PRINT TAB(10)"..↑£1/1///1///10&%.."  <059>
20009 PRINT TAB(10)"↑I..-.-...-...-,.,&%"  <005>
20010 PRINT TAB(10)".,.+/////////////2,.."  <093>
20011 PRINT TAB(10)".,+...............Z.."  <212>
20012 PRINT TAB(10)".+.................2."  <150>
20013 PRINT TAB(10)"+..................2{G
      REY 2}"                              <085>
20014 IF AP=50 THEN PRINT"{BROWN,HOME,7DOW
      N,16RIGHT}J{6RIGHT}J{4RIGHT}J"       <115>
20015 IF AP=19 OR AP=32 OR AP=55 THEN 2001
      7:REM TREPPE                         <142>
20016 GOTO 20020                           <048>
20017 POKE V+21,PEEK(V+21)OR 128:POKE 5017
      5,35:POKE V+14,160:POKE V+15,145     <159>
20018 POKE V+29,PEEK(V+29)OR 128:POKE V+46
      ,0:IF AP=19 THEN 19130               <254>
20020 IF AP<>33 AND AP<>56 AND AP<>58 THEN
       20900                               <037>
20021 POKE V+21,PEEK(V+21)OR 128:POKE 5017
      5,45:POKE V+14,159:POKE V+15,97      <027>
20022 POKE V+46,0:POKE V+23,PEEK(V+23)OR 1
      28:POKE V+29,PEEK(V+29)OR 129        <163>
20023 PRINT"{BROWN,HOME,6DOWN,17RIGHT}....
      ..{DOWN,6LEFT}.+..2.{DOWN,6LEFT}+...
      .2{DOWN,6LEFT}"                      <125>
20024 PRINT"{UP,17RIGHT}......{DOWN,6LEFT}
      ......{DOWN,6LEFT}"                  <234>
20900 IF AP=10 THEN 21001                  <182>
21000 IF PN(AP,1)=0 THEN 21005         0   <003>
21001 POKE 49363,55:POKE 49364,56:POKE 494
      03,52:POKE 49404,48:POKE 49443,54:PO
      KE 49444,48                          <065>
21002 POKE 49483,68:POKE 49484,44:POKE 495
      23,70:POKE 49524,57:POKE 49563,53:PO
      KE 49564,44                          <155>
21003 POKE 49603,27:POKE 49604,71:POKE 496
      43,68:POKE 49644,46:POKE 49683,51    <241>
21005 IF PN(AP,2)=0 THEN 21010             <248>
21006 POKE 49379,69:POKE 49380,65:POKE 494
      19,49:POKE 49420,29:POKE 49459,49:PO
      KE 49460,54                          <026>
21007 POKE 49499,45:POKE 49500,68:POKE 495
      39,64:POKE 49540,62:POKE 49579,45:PO
      KE 49580,61                          <231>
21008 POKE 49619,72:POKE 49620,60:POKE 496
      59,46:POKE 49660,68:POKE 49700,59    <228>
21010 IF PN(AP,3)=0 THEN 21030             <003>
21013 POKE 49410,36:POKE 49411,67:POKE 494
      12,67:POKE 49413,39:POKE 49449,48:PO
      KE 49450,44                          <208>
21014 POKE 49451,32:POKE 49452,32:POKE 494
      53,45:POKE 49454,49:POKE 49489,44:PO
      KE 49490,44                          <014>
21015 POKE 49491,32:POKE 49492,32:POKE 494
      93,45:POKE 49529,48:POKE 49530,44:PO
      KE 49414,45                          <006>
21016 POKE 49531,100:POKE 49532,103:POKE 4
      9533,45:POKE 49534,49:POKE 49569,44:
      POKE 49570,43                        <002>
21017 POKE 49571,46:POKE 49572,46:POKE 495
      73,50:POKE 49574,45:POKE 49610,46:PO
      KE 49611,46                          <183>
21018 POKE 49612,46:POKE 49613,46:POKE 494
      09,44                                <048>
21030 Z1=0:IF IN(10,1)=AP OR IN(23,1)=AP O
      R IN(26,1)=AP OR IN(28,1)=AP THEN 21
      034                                  <230>
21031 IF IN(35,1)=AP OR IN(36,1)=AP THEN 2
      1034                                 <252>
21032 IF IN(13,1)=AP OR IN(17,1)=AP OR IN(
      34,1)=AP OR IN(42,1)=AP THEN Z1=1:GO
      TO 21036                             <182>
21033 GOTO 21050                           <002>
21034 PRINT"{HOME,7DOWN,18RIGHT,4SPACE,DOW
      N,4LEFT,4SPACE,DOWN,4LEFT,4SPACE,DOW
      N,4LEFT,4SPACE}"                     <067>
21035 POKE V+10,161:POKE V+11,97:POKE V+29
      ,PEEK(V+29)OR 32                     <175>
21036 POKE V+21,PEEK(V+21)OR 32:POKE 50173
      ,39:POKE V+44,11:POKE V+23,PEEK(V+23
      )OR 32                               <198>
21037 IF Z1=1 THEN POKE V+10,161:POKE V+11
      ,78:POKE V+29,PEEK(V+29)OR 32        <133>
21038 IF Z1=1 THEN PRINT"{HOME,4DOWN,18RIG
      HT,4SPACE,DOWN,4LEFT,4SPACE,DOWN,4LE
      FT}";                                <089>
21039 IF Z1=1 THEN PRINT"{4SPACE,DOWN,4LEF
      T,4SPACE,DOWN,4LEFT,4SPACE}"         <063>
21050 IF IN(1,1)<>AP THEN 21060            <137>
21051 POKE V+21,PEEK(V+21)OR 1:POKE 50168,
      26:POKE V+0,200:POKE V+1,120:POKE V+
```

Listing »Grab des Pharao« (Fortsetzung)

```
      39,11                                 <190>
21052 POKE V+23,PEEK(V+23)OR 1:POKE V+29,P
      EEK(V+29)OR 1                         <147>
21060 IF IN(2,1)<>AP THEN 21070             <215>
21061 POKE V+21,PEEK(V+21)OR 128:POKE 5017
      5,29:POKE V+14,150:POKE V+15,110      <076>
21062 POKE V+46,0:POKE V+37,9:POKE V+38,1:
      POKE V+23,PEEK(V+23)OR 128:POKE V+28
      ,128                                  <139>
21063 POKE V+29,PEEK(V+29)OR 128            <112>
21070 IF IN(3,1)<>AP THEN 21080             <037>
21071 POKE V+21,PEEK(V+21)OR 64:POKE 50174
      ,28:POKE V+12,160:POKE V+13,92        <035>
21072 POKE V+45,15:POKE V+29,PEEK(V+29)OR
      64:POKE V+23,PEEK(V+23)OR 64          <009>
21080 IF IN(4,1)<>AP THEN 21090             <116>
21081 POKE V+21,PEEK(V+21)OR 128:POKE 5017
      5,30:POKE V+14,168:POKE V+15,108      <130>
21082 POKE V+46,9                           <026>
21090 IF IN(5,1)<>AP THEN 21100             <156>
21091 IF IN(14,1)<>-1 AND IN(14,1)<>35 THE
      N 21095                               <088>
21092 POKE V+21,PEEK(V+21)OR 128:POKE 5017
      5,53:POKE V+46,5:POKE V+14,195        <091>
21093 POKE V+15,130:POKE V+29,PEEK(V+29)OR
       128:GOTO 21100                       <049>
21095 X8=1:POKE V+21,PEEK(V+21)OR 128:POKE
       50175,46:POKE V+46,3                 <232>
21096 POKE V+14,190:POKE V+15,140:POKE V+2
      9,128:GOTO 164                        <079>
21100 IF IN(6,1)<>AP THEN GOTO 21110        <161>
21101 POKE V+21,PEEK(V+21)OR 8:POKE 50171,
      33:POKE V+28,8:POKE V+6,141:POKE V+7
      ,103                                  <074>
21102 POKE V+42,11:POKE V+37,0:POKE V+38,1
      5                                     <128>
21110 IF IN(7,1)<>AP THEN 21120             <056>
21115 PRINT"{HOME,BROWN,8DOWN,15RIGHT},H{D
      OWN,2LEFT}10{DOWN,2LEFT}-,{DOWN,2LEF
      T}-,{DOWN,2LEFT}//"                   <202>
21120 IF IN(9,1)<>AP THEN 21130             <199>
21121 IF AP=19 OR AP=51 THEN 21130          <003>
21122 POKE V+21,PEEK(V+21)OR 2:POKE 50169,
      34:POKE V+2,150:POKE V+3,140          <192>
21123 POKE V+29,PEEK(V+29)OR 2:POKE V+40,0 <111>
21130 IF IN(11,1)<>AP THEN 21140            <104>
21131 POKE V+21,PEEK(V+21)OR 64:POKE 50174
      ,37:POKE V+12,235:POKE V+13,122       <235>
21132 POKE V+28,64:POKE V+37,0:POKE V+38,9
      :POKE V+23,PEEK(V+23)OR 64            <233>
21140 IF IN(12,1)<>AP THEN 21150            <251>
21141 POKE V+21,PEEK(V+21)OR 64:POKE 50174
      ,38:POKE V+12,111:POKE V+13,119       <152>
21142 POKE V+28,64:POKE V+37,0:POKE V+38,9
      :POKE V+23,PEEK(V+23)OR 64            <243>
21150 IF IN(14,1)<>AP THEN 21160            <014>
21151 POKE V+21,PEEK(V+21)OR 4:POKE 50170,
      32:POKE V+4,170:POKE V+5,130          <150>
21152 POKE V+41,12:POKE V+29,PEEK(V+29)OR
      4:POKE V+23,PEEK(V+23)OR 4            <220>
21160 IF IN(15,1)<>AP THEN 21170            <160>
21161 POKE V+21,PEEK(V+21)OR 128:POKE 5017
      5,36:POKE V+14,170:POKE V+15,143      <077>
21162 POKE V+46,0:POKE V+29,PEEK(V+29)OR 1
      28                                    <090>
21170 IF IN(16,1)<>AP THEN 21180            <051>
21171 PRINT"{HOME,7DOWN}";TAB(24)"P"        <035>
21180 IF IN(18,1)<>AP AND IN(19,1)<>AP THE
      N 21200:REM SPRUENGE I.D. DECKE       <145>
21181 POKE V+21,PEEK(V+21)OR 128:POKE 5017
      5,43:POKE V+14,170:POKE V+15,56       <008>
21182 POKE V+46,0:POKE V+29,PEEK(V+29)OR 1
      28                                    <110>
21200 IF IN(20,1)<>AP THEN 21210            <090>
21201 IF AP=31 THEN POKE 55599,8:GOTO 2121
      0                                     <189>
21202 POKE V+21,PEEK(V+21)OR 8:POKE 50171,
      42:POKE V+6,180:POKE V+7,150:POKE V+
      42,8                                  <123>
21210 IF IN(21,1)<>AP THEN 21220            <236>
21211 POKE V+21,PEEK(V+21)OR 128:IF IN(46,
      3)=0 THEN POKE V+14,150:POKE V+15,12
      1:GOTO 21213                          <025>
21212 POKE V+14,155:POKE V+15,115           <117>
21213 POKE 50175,44:POKE V+46,9:POKE V+37,
      0:POKE V+38,1:POKE V+23,PEEK(V+23)OR
       128                                  <006>
21214 POKE V+29,PEEK(V+29)OR 128:POKE 4920
      9,50:POKE 49212,50:POKE 49250,47:POK
      E V+28,128                            <059>
21215 POKE 49251,47:POKE 49252,47:POKE 492
      10,45                                 <001>
21220 IF IN(22,1)<>AP THEN 21230            <127>
21221 POKE V+21,PEEK(V+21)OR 128:POKE 5017
      5,41                                  <164>
21222 POKE V+14,190:POKE V+15,105:POKE V+4
      5,14:POKE V+46,7                      <207>
21223 POKE V+23,PEEK(V+23)OR 128:POKE V+29
      ,PEEK(V+29)OR 128                     <197>
21230 IF IN(24,1)<>AP THEN 21240            <146>
21231 PRINT"{HOME,12DOWN,16RIGHT,GREY 1}1/
      1///10{DOWN,8LEFT}-.-...-,{DOWN,8LEF
      T}"                                   <053>
21232 PRINT"{UP,16RIGHT,BROWN}ROOOOOOE"     <142>
21240 IF IN(25,1)<>AP THEN 21250            <036>
21241 POKE V+21,PEEK(V+21)OR 192:POKE 5017
      4,48:POKE 50175,49:POKE V+45,7:POKE
      V+46,7:POKE V+12,138                  <151>
21242 POKE V+37,0:POKE V+38,12:POKE V+13,9
      8:POKE V+15,98:POKE V+28,192:POKE V+
      23,PEEK(V+23)OR 192                   <009>
21243 IF Y2=1 THEN POKE V+12,135:POKE V+14
      ,208:POKE V+29,32:POKE 50173,31:POKE
       V+44,9:GOTO 21247                    <084>
21244 PRINT"{HOME,6DOWN,14RIGHT,GREY 1}...
      .........{DOWN,12LEFT}............":
      POKE V+14,186                         <244>
21245 PRINT"{14RIGHT}............{DOWN,12L
      EFT}..........."                      <119>
21246 POKE V+29,PEEK(V+29)OR 192:PRINT"{14
      RIGHT}............":GOTO 21250        <159>
21247 PRINT"{HOME,6DOWN,14RIGHT,GREY 1}...
      {6SPACE}...{DOWN,12LEFT}...{6SPACE}.
      .."                                   <215>
21248 PRINT"{14RIGHT}...{6SPACE}...{DOWN,1
      2LEFT}...{6SPACE}..."                 <103>
21249 PRINT"{14RIGHT}...{6SPACE}..."        <078>
21250 IF IN(27,1)<>AP THEN 21260            <057>
21251 POKE V+21,PEEK(V+21)OR 128:POKE 5017
      5,51:POKE V+46,12:POKE V+14,200       <092>
21252 POKE V+15,120:POKE V+23,PEEK(V+23)OR
       128:POKE V+29,PEEK(V+29)OR 128       <042>
21260 IF IN(29,1)<>AP THEN 21270            <076>
21261 POKE V+21,PEEK(V+21)OR 192:POKE 5017
      4,52:POKE 50175,54                    <044>
21262 POKE V+12,121:POKE V+13,105:POKE V+1
      4,210:POKE V+15,105:POKE V+28,64      <043>
21264 POKE V+45,8:POKE V+46,15:POKE V+37,1
      1:POKE V+38,7                         <093>
21265 POKE V+23,PEEK(V+23)OR 192:POKE V+29
      ,PEEK(V+29)OR 64                      <136>
21270 IF IN(30,1)<>AP THEN 21280            <026>
21272 POKE V+21,PEEK(V+21)OR 24:POKE 50171
      ,50:POKE 50172,50:POKE V+6,142:POKE
      V+7,140                               <116>
21273 POKE V+8,172:POKE V+9,140:POKE V+42,
      8:POKE V+43,8:POKE V+29,PEEK(V+29)OR
       24                                   <231>
21280 IF IN(33,1)<>AP THEN 21290            <195>
21281 PRINT"{BROWN,HOME,DOWN,16RIGHT}2-...
      ..,+{DOWN,7LEFT}//////"               <117>
21290 IF IN(37,1)<>AP THEN 21300            <117>
21291 POKE V+21,PEEK(V+21)OR 128:POKE 5017
      5,58:POKE V+46,12:POKE V+14,138       <152>
21292 POKE V+15,105:POKE V+23,PEEK(V+23)OR
       128:POKE V+29,PEEK(V+29)OR 128       <098>
21300 IF IN(38,1)<>AP THEN 21310            <008>
21301 POKE V+21,PEEK(V+21)OR 208:POKE V+28
      ,64:POKE 50175,55:POKE 50172,57       <252>
21302 POKE 50174,56:POKE V+12,160:POKE V+1
      3,125:POKE V+14,160:POKE V+15,125:PO
      KE V+46,12                            <188>
21303 POKE V+9,125:POKE V+45,9:POKE V+37,0 <241>
21304 POKE V+43,12:POKE V+23,208:POKE V+29
      ,PEEK(V+29)OR 208                     <009>
21305 POKE V+8,160:IF IN(45,1)=-1 OR Y3=1
      THEN POKE V+8,180                     <252>
21307 PRINT"{HOME,10DOWN,19RIGHT,2SPACE,DO
      WN,2LEFT,2SPACE,DOWN,2LEFT,2SPACE,DO
      WN,3LEFT,4SPACE}"                     <250>
21310 IF IN(39,1)<>AP THEN 21320            <154>
21311 POKE V+21,PEEK(V+21)OR 128:POKE 5017
      5,30:POKE V+46,9:POKE V+14,172        <054>
21312 POKE V+15,98:POKE V+23,PEEK(V+23)OR
      128                                   <067>
21314 PRINT"{HOME,6DOWN,18RIGHT,BLACK}1..0
```




```
        {DOWN,4LEFT}-..,{DOWN,4LEFT}-..,"      <169>
21315 PRINT"{18RIGHT}-..,{DOWN,4LEFT}-..,"  <184>
21318 PRINT"{HOME,14DOWN,16RIGHT,BROWN}+1/
      ///02"                                  <213>
21320 IF IN(40,1)<>AP THEN 21330              <105>
21321 POKE V+21,PEEK(V+21)OR 128:POKE 5017
      5,52:POKE V+28,128:POKE V+46,8:POKE
      V+37,11                                 <079>
21322 POKE V+38,7:POKE V+14,160:POKE V+15,
      110:POKE V+23,PEEK(V+23)OR 128          <071>
21323 POKE V+29,PEEK(V+29)OR 128              <118>
21330 IF IN(44,1)<>AP THEN 21340              <125>
21331 PRINT"{HOME,12DOWN,15RIGHT,BROWN}±"     <247>
21340 IF IN(45,1)<>AP THEN 21350              <015>
21341 POKE V+21,PEEK(V+21)OR 32:POKE V+28,
      PEEK(V+28)OR 32:POKE V+44,14:POKE V+
      37,0                                    <058>
21342 POKE V+38,7:POKE V+29,PEEK(V+29)OR 3
      2:POKE V+10,160:POKE V+11,122:POKE 5
      0173,40                                 <063>
21350 IF IN(46,1)<>AP THEN 21370:REM GASMA
      SKE                                     <140>
21360 PRINT"{HOME,12DOWN,19RIGHT,BROWN}R"     <192>
21370 IF IN(41,1)<>AP THEN 21500              <011>
21372 IF AP=57 THEN 21500                     <030>
21375 PRINT"{HOME,13DOWN,20RIGHT,BROWN}T"     <150>
21500 RETURN                                  <220>
30000 POKE V+21,0:PRINT"{CLR,3DOWN,10RIGHT
      ,GREEN}SIE TRAGEN BEI SICH"             <187>
30010 FOR I=1 TO 46                           <062>
30020 IF IN(I,1)=-1 THEN PRINT"{BLUE,DOWN,
      10RIGHT,WHITE}"IT$(I)                   <170>
30025 NEXT                                    <061>
30026 PRINT"{5DOWN,10RIGHT,GREEN,RVSON}SPA
      CE DRUECKEN{RVOFF,WHITE}"               <235>
30030 GET A$:IF A$<>" "THEN 30030             <133>
30040 IF A$=" "THEN GOTO 160                  <221>
34000 POKE V+21,0:PRINT CHR$(14),"{CLR,3DO
      WN}":INPUT"FILENAME";E$                 <199>
34005 OPEN 1,8,15,"S:"+E$:CLOSE 1             <214>
34010 OPEN 2,8,2,E$+",W"                      <007>
34020 FOR I=1 TO 58                           <137>
34030 PRINT#2,PN(I,1):PRINT#2,PN(I,2):PRIN
      T#2,PN(I,3):PRINT#2,PN(I,4)             <216>
34035 PRINT#2,PN(I,5):PRINT#2,PN(I,6):NEXT    <191>
34040 FOR I=1 TO 46                           <027>
34050 PRINT#2,IN(I,1):PRINT#2,IN(I,2):PRIN
      T#2,IN(I,3):NEXT                        <158>
34060 PRINT#2,AP:PRINT#2,X1:PRINT#2,X2:PRI
      NT#2,X4:PRINT#2,X5:PRINT#2,X6           <074>
34070 PRINT#2,X9:PRINT#2,Y1:PRINT#2,Y2:PRI
      NT#2,Y3:PRINT#2,Y4:PRINT#2,GE           <067>
34080 CLOSE 2:GOTO 50300                      <210>
35000 PRINT"{CLR,3DOWN}":INPUT"FILENAME";E
      $:KK=1                                  <046>
35005 OPEN 2,8,2,E$                           <016>
35010 FOR I=1 TO 58                           <111>
35020 INPUT#2,PN(I,1),PN(I,2),PN(I,3),PN(I
      ,4),PN(I,5),PN(I,6):NEXT                <117>
35030 FOR I=1 TO 46                           <001>
35040 INPUT#2,IN(I,1),IN(I,2),IN(I,3):NEXT    <083>
35050 INPUT#2,AP,X1,X2,X4,X5,X6,X9,Y1,Y2,Y
      3,Y4,GE                                 <203>
35060 CLOSE 2:RETURN                          <140>
39900 DATA GEHE,ZERSTOERE,OEFFNE,KLOPFE,FR
      AGE,HACKE,TOETE,KRIECHE,WERFE,SCHIEB
      E                                       <117>
39910 DATA DREHE,KLETTERE,NIMM,VERLIERE,ZI
      EHE,LIST,SAVE,STOP                      <135>
39920 DATA SPITZHACKE,BEDUINE,STRAUCH,ZUGE
      MAUERTER EINGANG,SCHLANGE,STATUE,HEB
      EL                                      <081>
39925 DATA TUER NACH WESTEN,SEIL,TUER NACH
       NORDEN,DURCHGANG,DURCHGANG             <045>
39930 DATA TUER NACH NORDEN,TONKRUG MIT ES
      SEN,SKORPION,PFEIL,TUER NACH NORDEN     <004>
39935 DATA BESCHAEDIGTE DECKE,SPRUENGE IN
      DER DECKE,DIAMANT,TOTER GRABRAEUBER     <253>
39940 DATA STATUE,TUER NACH NORDEN,LOCH,RE
      LIEF,TUER NACH NORDEN,HORUS             <163>
39945 DATA TUER NACH NORDEN,SCHAETZE,HOLZB
      ALKEN,TUER NACH WESTEN,TUER             <130>
39950 DATA OEFFNUNG IN DER DECKE,TUER NACH
       NORDEN,TUER NACH NORDEN                <082>
39955 DATA TUER NACH NORDEN,OSIRISSTATUE,S
      ARKOPHAG,ZUGEMAUERTER AUSGANG           <134>
39960 DATA HOLZSCHREIN,AMULETT,TUER NACH N
      ORDEN,TUER NACH SUEDEN,SCHLUESSEL       <150>
39970 DATA TOTENMASKE,GASMASKE,GIFTIGE GAS
      E,FALLTUER,SCHRITTE,SONNENAUFGANG       <101>
39975 DATA GEHEIMGANG,AUGE,WAND,TREPPE        <186>
39985 DATA 160,0,162,0,142,32,208,142,33,2
      08,232,234,224,16,208,244,200,192,25
      5                                       <211>
39990 DATA 208,237,169,0,141,32,208,141,33
      ,208,96                                 <003>
40000 DATA 3,4,2,0,0,0,2,2,2,1,0,0,3,1,3,0
      ,0,0,1,0,5,0,0,0,0,0,0,4,0,0            <159>
40001 DATA 5,0,7,0,0,0,0,0,0,6,0,0,9,17,15
      ,0,0,0,10,8,0,0,0,0,0,9,0,0,0,0         <238>
40002 DATA 0,10,12,0,0,0,0,13,0,11,0,0,12,
      0,0,0,0,0,0,20,0,0,0,0,0,0,8,0,0        <082>
40003 DATA 0,0,0,15,0,0,8,0,18,0,0,0,0,0,1
      9,17,0,0,0,0,0,18,0,58,0,0,21,14,0,0    <009>
40004 DATA 0,0,0,20,0,0,27,23,0,21,0,0,22,
      24,0,26,0,0,23,0,0,0,0,0,26,0,24,0,0
      ,0                                      <046>
40005 DATA 0,25,23,0,0,0                      <044>
40006 DATA 0,22,0,28,0,0,31,0,27,29,0,0,0,
      0,28,0,0,0,0,0,0,0,0,0,28,0,30,0,0      <098>
40007 DATA 0,0,0,31,0,33,34,38,0,36,32,0,0
      ,33,0,35,0,0,55,36,34,0,0,0,35,37,33
      ,0                                      <134>
40008 DATA 0,0,36,41,0,0,0,0,33,0,0,0,0,0,
      0,40,46,0,0,0,39,0,0,0,0,0,37,42,0,0
      ,0                                      <021>
40009 DATA 0,41,0,0,0,0,0,0,44,48,0,0,0,43
      ,50,49,0,0,0,49,0,0,0,0,0,0,0,51,39     <240>
40010 DATA 0,0,0,0,0,40,0,0,0,49,0,43,0,0,
      48,45,0,44,0,0,44,0,0,38,0,0,0,0,0,4
      6,0                                     <099>
40011 DATA 0,0,0,0,47,0,0,0,0,0,52,0,0,0,0
      ,0,51,0,0,0,35,0,0,0,56,0,57,0,0,55,
      0                                       <025>
40012 DATA 56,0,0,0,0,0,0,0,0,0,19,0          <192>
40025 DATA 4,1,1,5,0,1,7,0,1                  <230>
40027 DATA 7,0,0,35,0,1,10,1,1                <169>
40029 DATA 10,0,0,10,0,1,12,1,1               <170>
40030 DATA 13,0,1,0,0,0                       <070>
40031 DATA 14,0,1,15,0,1,16,1,1,16,0,0        <209>
40032 DATA 21,0,0                             <034>
40033 DATA 21,0,1,23,0,1                      <072>
40034 DATA 24,0,1,25,1,1                      <169>
40035 DATA 27,1,1,31,0,1,31,0,1               <105>
40036 DATA 37,0,1                             <143>
40037 DATA 40,0,1,42,0,1,42,0,1               <221>
40038 DATA 38,0,1,47,0,0,48                   <225>
40039 DATA 1,1,50,0,1,50,0,0                  <039>
40040 DATA 51,0,1,52,0,1,0,0,0                <240>
40041 DATA 47,0,1,52,0,1,53,0,1               <091>
40043 DATA 54,0,1,57,0,1,0,1,1                <115>
40044 DATA 19,0,1,25,0,1,58,1,1,53,1,0,27,
      1,0                                     <035>
45000 POKE 53270,PEEK(53270)OR 16:POKE 532
      83,9:POKE 53282,7                       <054>
45005 POKE 53280,0:POKE 53281,9               <208>
45010 PRINT"{CLR,GREY 1,2DOWN,3RIGHT,4SPAC
      E}P H A R A O S{3SPACE}G R A V E"       <176>
45020 PRINT"{DOWN,11SPACE}A D V E N T U R
      E{13SPACE,BLACK}"                       <005>
45030 PRINT"{19SPACE}.."                      <099>
45031 PRINT"{18SPACE}.//."                    <102>
45032 PRINT"{17SPACE}.////."                  <114>
45033 PRINT"{16SPACE}.//////."                <162>
45034 PRINT"{15SPACE}.////////."              <098>
45035 PRINT"{14SPACE}.//////////."            <101>
45036 PRINT"{13SPACE}.////////////."          <113>
45037 PRINT"{12SPACE}.//////////////."        <161>
45038 PRINT"{11SPACE}.////////////////."      <097>
45039 PRINT"{10SPACE}.//////////////////."    <100>
45040 PRINT"{9SPACE}.////////////////////."   <112>
45041 PRINT"{8SPACE}.//////////////////////
      /."                                     <160>
45050 PRINT"{2DOWN,GREY 1,19SPACE}BY{19SPA
      CE}"                                    <166>
45060 PRINT"{12SPACE}WOLFGANG RAUSCH{12SPA
      CE}"                                    <082>
45100 GOSUB 12000:GOSUB 30                    <059>
45110 PRINT"{CLR}",CHR$(14):POKE 53270,PEE
      K(53270)AND 239                         <192>
45112 PRINT"{WHITE,3DOWN,4RIGHT}N : NEUES
      SPIEL BEGINNEN"                         <114>
```

Listing »Grab des Pharao« (Fortsetzung)

```
45113 PRINT"{DOWN,4RIGHT}A : ALTES SPIEL F
      ORTSETZEN"                          <100>
45114 GET X$:IF X$="N"THEN DA=1:GOTO 45120 <138>
45116 IF X$="A"THEN GOSUB 35000:GOSUB 60:G
      OTO 45120                           <136>
45118 IF X$<>"A"AND X$<>"N"THEN 45114     <174>
45120 PRINT"{CLR,RIGHT,YELLOW}ZIEL DES SPI
      ELS IST ES EINE PYRAMIDE"           <118>
45125 PRINT"{RIGHT}ZU ERFORSCHEN UND DIE W
      ERTVOLLE"                           <042>
45127 PRINT"{RIGHT}TOTENMASKE DES PHARAOS
      ZU FINDEN."                         <077>
45130 PRINT"{DOWN,RIGHT,ORANGE}FOLGENDE VE
      RBEN VERSTEHT DER COMPUTER:"        <187>
45132 PRINT"{RIGHT,DOWN}GEHE,NIMM,VERLIERE
      ,OEFFNE,TOETE"                      <007>
45134 PRINT"{RIGHT}ZERSTOERE,KLOPFE,FRAGE,
      WERFE,DREHE"                        <163>
45136 PRINT"{RIGHT}HACKE,KRIECHE,WERFE,SCH
      IEBE,KLETTERE"                      <157>
45137 PRINT"{RIGHT}LIST,SAVE,STOP         <117>
45140 PRINT"{BLACK,DOWN,RIGHT}NACH JEDEM E
      INGEGEBENEN WORT MUSS"              <150>
45142 PRINT"{RIGHT}'RETURN' GEDRUECKT WERD
      EN."                                <237>
45144 PRINT"{RIGHT}ALLE NOMEN KOENNEN AUF
      5,ALLE VERBEN"                      <143>
45146 PRINT"{RIGHT}AUF 4 BUCHSTABEN ABGEKU
      ERZT WERDEN."                       <188>
45148 PRINT"{RIGHT}BEI RICHTUNGSANGABEN GE
      NUEGT EIN "                         <090>
45150 PRINT"{RIGHT}BUCHSTABE.             <245>
45152 PRINT"{RIGHT}Z.B.:GEHE 'RETURN'{5SPA
      CE}N 'RETURN'"                      <084>
45154 PRINT"{RIGHT}LIST ZEIGT ALLE GEGENST
      AENDE, DIE"                         <162>
45155 PRINT"{RIGHT}MAN BEI SICH TRAEGT (MA
      XIMAL 3)."                          <037>
45157 PRINT"{RIGHT}SAVE SPEICHERT DEN SPIE
      LSTAND AUF"                         <247>
45158 PRINT"{RIGHT}KASSETTE AB."          <150>
45160 IF DA=1 THEN GOSUB 60               <091>
45170 PRINT"{DOWN,4RIGHT,RVSON}'SPACE' DRU
      ECKEN{RVOFF}"                       <050>
47000 GET X$:IF X$<>" "THEN 47000         <086>
47010 IF X$=" "THEN PRINT CHR$(142),"{CLR}
      "                                   <244>
47020 IF KK=1 THEN 155                    <059>
47025 GOTO 150                            <040>
49000 GOSUB 5600:PRINT"{CLR,2DOWN,9RIGHT}S
      IE SIND AUSSERHALB":GOSUB 5800      <184>
49002 IF IN(45,1)<>-1 THEN 49050          <129>
49005 PRINT CHR$(14),"{CLR,GREEN,3DOWN,3SP
      ACE}SIE HABEN ES GESCHAFFT DIE GOLDE
      NE"                                 <247>
49010 PRINT"{DOWN,3SPACE}TOTENMASKE DES PH
      ARAOS ZU FINDEN"                    <054>
49020 PRINT"{DOWN,3SPACE}UND HOECHSTE EHRU
      NGEN SIND IHNEN{3SPACE}"            <204>
49030 PRINT"{DOWN,3SPACE}GEWISS,"         <048>
49040 PRINT"{3DOWN,RED,3SPACE}DOCH DER FLU
      CH DES PHARAOS WIRD"                <226>
49045 PRINT"{DOWN,3SPACE}SIE FUER IMMER VE
      RFOLGEN.{WHITE}":GOTO 50300         <104>
49050 PRINT"{CLR,GREEN,3DOWN,3SPACE}SIE HA
      BEN ES ZWAR GESCHAFFT, DIE"         <254>
49055 PRINT"{DOWN,3SPACE}PYRAMIDE LEBEND Z
      U VERLASSEN, DOCH"                  <019>
49060 PRINT"{DOWN,3SPACE}SIE HABEN DIE GOL
      DENE TOTENMASKE"                    <242>
49065 PRINT"{DOWN,3SPACE}NICHT GEFUNDEN.":
      GOTO 50300                          <156>
50100 REM                                 <121>
50105 POKE V+14,190:POKE V+15,130         <233>
50110 FOR F2=135 TO 155 STEP 2:POKE 50175,
      46:FOR T=1 TO 100:NEXT:POKE V+15,F2 <142>
50120 POKE 50175,47:FOR T=1 TO 100:NEXT:NE
      XT                                  <022>
50121 FOR T=1 TO 30:SYS 866:NEXT:GOTO 5021
      0                                   <217>
50190 PRINT CHR$(14)"{CLR,RED,2DOWN,3SPACE
      }SIE HABEN SICH HOFFNUNGSLOS VERIRRT
      {WHITE}":GOTO 50300                 <022>
50200 PRINT CHR$(14)"{CLR,RED,2DOWN,3SPACE
      }SIE HABEN DEN HALT VERLOREN UND SIN
      D{4SPACE}HINUNTERGEFALLEN{WHITE}"   <050>
50201 GOTO 50300                          <254>
50202 PRINT CHR$(14)"{CLR,RED,2DOWN,3SPACE
      }SIE SIND IN EINE FALLGRUBE GESTUERZ
      T{WHITE}":GOSUB 5600:GOTO 50300     <003>
50203 GOSUB 5600:PRINT CHR$(14)"{CLR,RED,2
      DOWN,3SPACE}DIE "IT$(54)" IST EINGES
      TUERZT{WHITE}":GOTO 50300           <209>
50204 GOSUB 5600:PRINT CHR$(14)"{RED,CLR,2
      DOWN,3SPACE}DIE DECKE IST HERUNTERGE
      BROCHEN{WHITE}":GOTO 50300          <114>
50206 PRINT CHR$(14)"{CLR,RED,2DOWN,3SPACE
      }EIN HERABFALLENDER STEIN HAT SIE ER
      {5SPACE}SCHLAGEN{WHITE}":GOTO 50300 <211>
50207 PRINT CHR$(14)"{CLR,RED,2DOWN,3SPACE
      }"IT$(47)" HABEN SIE GETOETET{WHITE}
      ":GOTO 50300                        <058>
50208 FOR T=1 TO 20:SYS 866:NEXT:GOSUB 560
      0:PRINT"{CLR,RED,2DOWN,3SPACE}SIE WU
      RDEN VON HINTEN GEPACKT UND "       <137>
50209 PRINT"{3SPACE}ERWUERGT{WHITE}":GOTO
      50300                               <177>
50210 GOSUB 5600:PRINT CHR$(14)"{CLR,RED,2
      DOWN,3SPACE}DIE "IT$(5)" HAT SIE GEB
      ISSEN{WHITE}":GOTO 50300            <131>
50211 GOSUB 5600:PRINT CHR$(14)"{CLR,RED,2
      DOWN,3SPACE}EINE "IT$(48)" HAT SICH
      UNTER IHNEN"                        <103>
50212 PRINT"{3SPACE}GEOEFFNET{WHITE}":GOTO
      50300                               <201>
50213 GOSUB 5600:PRINT CHR$(14)"{CLR,RED,2
      DOWN,3SPACE}SIE HABEN EINEN MECHANIS
      MUS "                               <138>
50214 PRINT"{3SPACE}AUSGELOEST ,WODURCH EI
      N "IT$(16)" AUS "                   <160>
50215 PRINT"{3SPACE}DER WAND GESCHOSSEN WU
      RDE{WHITE}":GOTO 50300              <099>
50300 PRINT CHR$(14)"{5DOWN,3RIGHT}NOCH EI
      N VERSUCH?{SHIFT-SPACE}(J/N)"       <148>
50310 GET A$:IF A$<>"J"AND A$<>"N"THEN 503
      10                                  <103>
50320 IF A$="J"THEN CLR:PRINT CHR$(142)"{C
      LR}":RUN                            <026>
50330 SYS 64738                           <095>
50700 FOR X=1 TO 20:POKE V+8,160+X:FOR T=1
       TO 200:NEXT:NEXT:IN(45,3)=1:GOSUB 5
      900:GOTO 164                        <192>
50720 TI$="000000"                        <058>
50725 PRINT A4$"KOMMANDO ?"               <062>
50730 POKE 631,0:POKE 198,0               <076>
50731 U=PEEK(631)                         <124>
50740 IF TI$="000002"THEN 50770           <008>
50750 IF U=71 THEN POKE 631,U:GOTO 250    <069>
50760 IF U>0 AND U<>71 THEN 50770         <060>
50765 GOTO 50731                          <246>
50770 FOR X=1 TO 10:POKE V+13,125-X:FOR T=
      1 TO 300:NEXT:NEXT:FOR T=1 TO 20:SYS
       866:NEXT                           <161>
50780 GOSUB 5600:PRINT CHR$(14)"{CLR}":GOT
      O 50300                             <086>
50800 TI$="000000":POKE 631,0             <005>
50805 PRINT A4$"KOMMANDO ?"               <142>
50806 POKE 631,0:POKE 198,0               <152>
50807 U=PEEK(631)                         <200>
50808 IF U=71 OR U=84 THEN POKE 631,U:GOTO
       250                                <081>
50809 IF U>0 AND U<>71 AND U<>84 THEN 5082
      0                                   <012>
50810 IF TI$="000002"THEN 50820           <042>
50815 GOTO 50807                          <140>
50820 FOR X=1 TO 10:POKE V+15,140+X:POKE V
      +14,170-X/2:FOR T=1 TO 100:NEXT:NEXT <174>
50825 FOR T=1 TO 10:SYS 866:NEXT:GOSUB 560
      0                                   <138>
50830 PRINT CHR$(14)"{CLR,RED,2DOWN,3SPACE
      }DER "IT$(15)" HAT SIE GESTOCHEN{WHI
      TE}":GOTO 50300                     <011>
52000 PRINT A3$"{GREEN}SIE HOEREN "IT$(49)
      " HINTER SICH{WHITE}":C=C+1         <095>
52004 FOR C1=1 TO 4                       <022>
52005 FOR LA=5 TO 0 STEP-.4               <018>
52010 POKE SI+24,LA:POKE SI+5,8*16+10:POKE
       SI+6,15*16:POKE SI+1,15:POKE SI+0,0 <060>
52020 POKE SI+4,129:NEXT                  <065>
52025 POKE SI+4,0:POKE SI+5,0             <236>
52030 FOR T=1 TO 800:NEXT:NEXT:RETURN     <103>
```


Listing »Grab des Pharao« (Schluß)

Castle of Doom

Castle of Doom ist ein Adventure, das es in sich hat. Man kann nicht nur drei verschiedene Versionen auswählen, es ist zudem gar nicht einfach, bis zum Ende durchzustehen. Dazu brauchen Sie Geduld, eine gehörige Portion Phantasie und — viel Zeit.

Zunächst einige wesentliche Informationen zum Ablauf des Spiels:

Am Anfang des Spiels kann man sich für eine von drei verschiedenen Versionen entscheiden. Bei allen Versionen ist das »Spielfeld« gleich, doch das Ziel, das es zu erreichen gilt, ist jeweils ein anderes. Die Kommandos müssen jeweils aus zwei Worten bestehen, die durch Leerraum getrennt werden. Beide Worte können beliebig abgekürzt werden, doch man muß darauf achten, daß keine Mißverständnisse auftreten.

Das Programm beinhaltet eine kurze Anleitung, in der die wichtigsten Punkte der Befehlseingabe nochmals erwähnt werden. Zudem sind alle Verben, die der Computer versteht, aufgeführt. Wird dennoch ein Wort benutzt, das der Computer nicht kennt, so gibt er an, welches der eingegebenen Worte ihm unbekannt ist.

Auf dem Bildschirm sind immer zu sehen:

— eine Grafik, die den Ort zeigt, an dem man sich befindet,

— eine Liste der Gegenstände, die sich an diesem Ort befinden,

— eine Liste der Gegenstände, die man besitzt.

(Bernd Weißbecker/gk)

Zum Abschluß hier noch die wichtigsten Variablen:

Variable	Bedeutung
V$(1...14)	Liste Verben
H$(1...21)	Liste Hauptwörter
RI$(1...19)	mögliche Richtungen
ZI$(1...19)	Nachbarfelder
G$(1...12)	Liste Gegenstände
G(1...12)	Ort der Gegenstände
P	Position
V	Version
Z1$,Z2$	Zauberwörter
ZA	= 0,1,2 Zauberwort gesagt?
K$	Kommando
V$	Verb
H$	Hauptwort
KB$(1...9)	mögliche Kombinationen
MO	= 1 Monster besiegt
GE	= 1 Geist besiegt
MR	= 1 Monster anwesend
GS	= 1 Geist anwesend
ZB	= 1 Zauberer anwesend
GG	Anzahl Gegenstände

Variablen-Definition zu Castle of Doom

Zur Erläuterung des Programms sind hier die verschiedenen Abschnitte aufgeführt:

Zeilen	Abschnitt
1	Sprung zur Befehlsausführung
5 — 95	DATAs einlesen, Erklärung?
100 — 160	Version auswählen
165 — 176	Zauberwörter bestimmen
179 — 740	Bilder zeichnen
748 — 860	Kommando auswerten
870 — 940	Bewegen
950 — 980	Sprung zur Befehlsausführung berechnen und ausführen
999 — 5382	Befehlsausführung
6500 — 6528	Befehl »sage«
7000 — 7065	Besitz + Gegenstände anzeigen
7100 — 7125	Befehl »nimm«
7200 — 7415	Befehl »wirf«
7500 — 7560	Erklärung
7600 — 7720	verloren
7800 — 7880	gewonnen
8000 — 8110	DATAs

Da die Kommandoauswertung recht ungewöhnlich ausgeführt wird, folgt hier noch eine Beschreibung der betreffenden Programmzeilen:

Zeilen	Beschreibung
755	Kommandoeingabe
760 — 785	Zerlegen in Verb/Hauptwort
795 — 810	Vergleich mit der Liste aller bekannten Wörter Zuordnung: X = Nummer des Verbs (1 — 14) Y = Nummer des Hauptworts (1 — 21)
830 — 845	Untersuchen, ob eine sinnvolle Kombination von Verb und Hauptwort vorliegt
960 — 980	Aus X und Y wird die Zeilennummer berechnet, in der die Antwort auf das eingegebene Kommando steht. Die berechnete Zeilennummer wird in ihre Ziffern zerlegt, und diese werden in den Programmspeicher hinter das GOTO in Zeile 1 gePOKEd. Anschließend wird ein GOTO 1 ausgeführt.

```
0 GOTO 5                                              <207>
1 GOTO!!!!                                            <174>
2 REM                                                 <064>
3 REM ZEILEN 0 UND 1 NICHT AENDERN !!!! .             <010>
4 REM                                                 <066>
5 DIM V$(14),H$(21),RI$(19),ZI$(19),G$(12)
  ,G(12)                                              <251>
6 L$="{RVSON,24SPACE}"                                <092>
7 CL$="G{22SPACE}H"                                   <083>
8 EF$="{39SPACE}"                                     <128>
10 POKE 53280,14:POKE 53281,14:POKE 53272,
   23:P=15:QQ=54272:VI=53248                          <212>
15 POKE VI+21,0:PRINT"{CLR}"SPC(12)"{BLACK
   ,8DOWN}CASTLE{SHIFT-SPACE}OF{SHIFT-SPAC
   E}DOOM":REM CASTLE OF DOOM                         <081>
20 PRINT"{2DOWN,WHITE,4SPACE}EIN ADVENTURE
    VON B.WEISSBECKER"                                <187>
25 PRINT SPC(11)"{2DOWN}* BITTE WARTEN *"             <138>
30 FOR I=1 TO 14:READ V$(I)                           <037>
35 IF I<10 THEN READ KB$(I)                           <023>
40 NEXT                                               <050>
45 FOR I=1 TO 21:READ H$(I):NEXT                      <129>
50 FOR I=1 TO 19:READ RI$(I),ZI$(I):NEXT              <087>
55 FOR I=1 TO 12:G$(I)=H$(I):READ G(I):NEX
   T                                                  <249>
60 FOR I=1 TO 4:READ ZX$(I),ZY$(I):NEXT               <187>
65 FOR I=13 TO 15                                     <155>
70 FOR J=0 TO 62:READ K:POKE 64*I+J,K:NEXT
   :NEXT                                              <224>
85 PRINT"{UP,2SPACE}MOECHTEN SIE EINE ERKL
   AERUNG ? (J/N)"                                    <198>
90 GET A$:IF A$="J"THEN 7500                          <101>
95 IF A$<>"N"THEN 90                                  <200>
100 PRINT"{CLR,BLACK}"SPC(10)"{DOWN}CASTLE
    {SHIFT-SPACE}OF{SHIFT-SPACE}DOOM"                 <198>
105 PRINT"{2DOWN,CYAN,SPACE}BEI DIESEM ADV
    ENTURE KOENNEN SIE ZWI-"                          <167>
110 PRINT"{DOWN,SPACE}SCHEN DREI VERSIONEN
     WAEHLEN :":PRINT"{DOWN,2SPACE}WOLLEN
    SIE :"                                            <208>
115 PRINT"{DOWN,2SPACE}1 - EINE VERZAUBERT
    E PRINZESSIN":PRINT SPC(6)"{DOWN}RETTE
    N"                                                <101>
120 PRINT"{DOWN,2SPACE}2 - EINEN BOESEN ZA
    UBERER BESIEGEN"                                  <042>
125 PRINT"{DOWN,2SPACE}3 - EINEN SCHATZ SU
    CHEN"                                             <049>
130 PRINT"{2DOWN,SPACE}BITTE GEBEN SIE DIE
     ENTSPRECHENDE"                                   <104>
135 PRINT"{DOWN,SPACE}KENNZIFFER EIN."                <073>
140 POKE 198,0:WAIT 198,1                             <112>
145 GET I$:IF ASC(I$)<49 OR ASC(I$)>51 THE
    N 140                                             <140>
150 V=VAL(I$):IF V<>2 THEN 160                        <243>
155 RI$(13)="W-O":ZI$(13)="1214":RI$(9)="W
    -O-R":ZI$(9)="081018"                             <175>
157 RI$(8)="N-S-O":ZI$(8)="051209"                    <054>
160 IF V=1 THEN G(7)=1                                <063>
165 FOR I=1 TO 4:J(I)=1+4*RND(TI):NEXT                <221>
170 Z1$=ZX$(J(1))+ZY$(J(2))                           <122>
175 Z2$=ZX$(J(3))+ZY$(J(4))                           <040>
176 IF Z1$=Z2$THEN 165                                <202>
177 TI$="000000"                                      <156>
179 REM **** BILDER ZEICHNEN ****                     <125>
180 PRINT"{CLR,BLACK}";:POKE 53272,21:POKE
     VI+21,0:POKE VI+23,0:POKE VI+29,0                <150>
185 FOR I=1 TO 20:PRINT SPC(8)CL$:NEXT                <027>
190 FOR I=1824 TO 1863:POKE I,99:POKE I+QQ
    ,0:NEXT                                           <075>
195 IF P>11 THEN 415                                  <061>
197 POKE 53280,5:POKE 53281,5                         <099>
200 PRINT"{HOME}";:FOR I=1 TO 4:PRINT SPC(
    8+I)"M"SPC(22-2*I)"N":NEXT                        <160>
205 PRINT SPC(13)"{UP}PPPPPPPPPPPPPP"                 <076>
210 FOR I=1 TO 10:PRINT SPC(12)"H"SPC(14)"
    G":NEXT                                           <151>
215 PRINT SPC(13)"YYYYYYYYYYYYYY{UP}"                 <231>
220 FOR I=1 TO 5:PRINT SPC(13-I)"N"SPC(12+
    2*I)"M":NEXT                                      <216>
225 FOR I=1 TO LEN(RI$(P))STEP 2:A$=MID$(R
    I$(P),I,1)                                        <037>
230 IF A$="N"THEN 250                                 <074>
235 IF A$="W"THEN 260                                 <219>
240 IF A$="O"THEN 265                                 <044>
245 NEXT:GOTO 270                                     <147>
250 FOR J=1402 TO 1562 STEP 40:FOR K=0 TO
    3:POKE J+K,160:POKE J+K+QQ,0:NEXT:NEXT            <093>
255 POKE 1402,254:POKE 1405,252:GOTO 245              <083>
260 FOR J=1552 TO 1752 STEP 40:POKE J,160:
    POKE QQ+J,0:NEXT:POKE 1752,105:GOTO 24
    5                                                 <194>
265 FOR J=1575 TO 1775 STEP 40:POKE J,160:
    POKE QQ+J,0:NEXT:POKE 1775,95:GOTO 245            <110>
270 ON P GOTO 275,740,300,315,740,325,345,
    740,355,375,395                                   <019>
275 ON V GOTO 740,280,285                             <081>
280 POKE 2040,15:POKE VI+39,0:POKE VI,150:
    POKE VI+1,140:POKE VI+29,1:POKE VI+23,
    1                                                 <134>
282 POKE VI+21,1:ZB=1:GOTO 740                        <172>
285 PRINT"{HOME,10DOWN}"SPC(13)"PPPP":PRIN
    T SPC(12)"N{3SPACE}NH"                            <074>
290 PRINT SPC(11)"N{3SPACE}N H":PRINT SPC(
    11)"OYYP{2SPACE}H":PRINT SPC(11)"H$*H
    N"                                                <022>
295 PRINT SPC(11)"LPPPN SCHATZ":FOR I=1 TO
     2000:NEXT:GOTO 7830                              <255>
300 IF LEFT$(RI$(3),1)="N"THEN 740                    <140>
305 FOR J=1402 TO 1562 STEP 40:FOR K=0 TO
    3                                                 <013>
310 POKE J+K,160:POKE J+K+QQ,06:NEXT:NEXT:
    PRINT SPC(16)"{4UP,BLACK}SPIEGEL":GOTO
     740                                              <061>
315 PRINT"{HOME,2DOWN}"SPC(13)"*{RVSON,4SP
    ACE,RVOFF}£":PRINT SPC(14)"*{RVSON,2SP
    ACE,RVOFF}£":PRINT SPC(15)"HN"                    <194>
320 FOR I=1 TO 10:PRINT SPC(15)"OP":NEXT:G
    OTO 740                                           <192>
325 PRINT"{HOME,BROWN,10DOWN}"SPC(13)"PPPP
    P":PRINT SPC(13)"LPPPG"                           <143>
330 PRINT SPC(12)"N{4SPACE}NH":PRINT SPC(1
    1)"N{4SPACE}NNH":PRINT SPC(10)"N{4SPAC
    E}NNY"                                            <133>
335 PRINT SPC(9)"OYYYPNN":PRINT SPC(9)"H{3
    SPACE}HN":PRINT SPC(9)"LPPPG{2SPACE}SC
    HLAFZIMMER"                                       <201>
340 GOTO 740                                          <126>
345 IF GE=1 THEN 740                                  <235>
350 POKE 2040,13:POKE VI+39,1:POKE VI,150:
    POKE VI+1,150:POKE VI+23,1:POKE VI+29,
    1                                                 <004>
351 POKE VI+21,1:IF G(12)<>0 THEN FOR I=1
    TO 2000:NEXT:GOTO 7630                            <008>
352 GS=1:GOTO 740                                     <027>
355 PRINT"{HOME,BLACK,DOWN}"SPC(11)"BURGHO
    F / BRUNNEN":PRINT SPC(13)"{DOWN}OLOLO
    LOPGPGPGP"                                        <220>
360 PRINT SPC(17)"{9DOWN}GPOOPG":PRINT SPC
    (16)"{RVSON,8SPACE}"                              <209>
365 PRINT SPC(16)"HTYUUYTN":PRINT SPC(16)"
    H{6SPACE}N":PRINT SPC(16)"YEDCCDEY"               <128>
370 GOTO 740                                          <156>
375 POKE 214,14:PRINT                                 <178>
380 IF FA=1 THEN PRINT SPC(23)"{DOWN}*{RVS
    ON,3SPACE}*":PRINT SPC(24)"*{RVSON,3SP
    ACE}*":GOTO 740                                   <098>
385 PRINT SPC(23)"PPPP":PRINT SPC(23)"M{3S
    PACE}M"                                           <215>
390 PRINT SPC(16)"QUADRAT M{3SPACE}M":PRIN
    T SPC(25)"YYYY":GOTO 740                          <179>
395 PRINT"{HOME,9DOWN}"SPC(23)"PPPP":PRINT
     SPC(22)"NM{3SPACE}M":PRINT SPC(22)"N
    OPYP"                                             <054>
400 PRINT SPC(15)"LG{5SPACE}N HN.N":PRINT
    SPC(15)"LG{5SPACE}N HN N"                         <021>
405 PRINT SPC(14)"{RVSON}£{2SPACE}*{RVOFF}
    YYYYYMLGPG":PRINT SPC(13)"{RVSON}£{4SP
    ACE}*"                                            <187>
410 PRINT SPC(21)"SCHRANK":GOTO 740                   <244>
415 IF P>14 THEN 445                                  <226>
417 POKE 53280,8:POKE 53281,8                         <083>
420 PRINT"{HOME}";:FOR I=1 TO 6:PRINT SPC(
    8)"{BLUE}"L$:NEXT                                 <177>
425 FOR I=55504 TO 55514 STEP 2:POKE I,7:N
    EXT:FOR I=55517 TO 55527 STEP 2:POKE I
    ,7:NEXT                                           <199>
```

Listing Castle of Doom. Bitte mit dem Checksummer auf Seite 16 eingeben.

```
430 FOR I=1 TO 11:PRINT"{YELLOW}"SPC(8)L$:
    NEXT:FOR I=1 TO 3:PRINT"{BROWN}"SPC(8)
    L$:NEXT                                <203>
435 IF LEFT$(RI$(P),1)<>"N"THEN 740        <170>
440 POKE 214,11:PRINT"{BLACK}":FOR I=1 TO
    5:PRINT SPC(18)"{RVSON,4SPACE}":NEXT   <016>
442 GOTO 740                               <228>
445 POKE 53280,14:POKE 53281,14:ON P-14 GO
    TO 555,450,485,500,540                 <013>
450 PRINT"{HOME,YELLOW,6DOWN}"SPC(20)"B":P
    RINT SPC(18)"{RVSON,3SPACE}"           <175>
460 PRINT SPC(17)"{BROWN}B{RVSON,SPACE}N C
    ":PRINT SPC(16)"{RVSON}£ N{3SPACE}C"   <048>
465 PRINT SPC(15)"B{RVSON,SPACE}({6SPACE}C
    ":PRINT SPC(14)"T{RVSON}V{2SPACE}M{7SP
    ACE}C"                                 <072>
470 PRINT SPC(13)"{RVSON}£{5SPACE}M{7SPACE
    }C"                                    <234>
475 FOR I=1 TO 7:PRINT SPC(8)L$:NEXT       <205>
480 PRINT SPC(9)"{6UP,RVSON}BERG"          <000>
481 IF LEN(RI$(16))=5 THEN PRINT SPC(9)"{D
    OWN,RVSON}STEILER WEG NACH OBEN"       <067>
482 GOTO 740                               <012>
485 PRINT"{HOME}"SPC(9)"{4DOWN}STEILER BER
    GHANG,":PRINT SPC(9)"{DOWN}HOEHLE"     <233>
490 PRINT SPC(21)"{3DOWN}T":PRINT SPC(19)"
    B{RVSON,3SPACE}VC{RVOFF}F"             <219>
495 FOR I=1 TO 6:PRINT SPC(19)"{RVSON,7SPA
    CE}":NEXT:GOTO 740                     <195>
500 PRINT"{HOME}";:FOR I=1 TO 19:PRINT SPC
    (8)L$:NEXT                             <052>
505 PRINT"{HOME}"SPC(9)"{RVSON,DOWN}IM BRU
    NNEN ..."                              <032>
510 IF V<>3 THEN 740                       <079>
515 IF MO=1 THEN PRINT"{DOWN}":GOTO 530    <036>
520 POKE 2040,14:POKE VI+39,13:POKE VI,200
    :POKE VI+1,150:POKE VI+23,1:POKE VI+29
    ,1                                     <244>
525 PRINT SPC(9)"{RVSON,DOWN}MONSTER":POKE
     VI+21,1:MR=1                          <226>
530 PRINT SPC(9)"{RVSON,DOWN}TUER{5DOWN}":
    FOR I=1 TO 7:PRINT SPC(16)"{5SPACE}":N
    EXT                                    <047>
535 IF RI$(18)="H"THEN POKE 1601,114       <170>
537 GOTO 740                               <069>
540 PRINT"{HOME}";:FOR I=1 TO 20:PRINT SPC
    (8)L$:NEXT                             <060>
545 PRINT"{HOME}"SPC(9)"{RVSON,DOWN}IN DER
     HOEHLE ..."                           <115>
550 PRINT SPC(9)"{RVSON,DOWN}STOCKDUNKEL H
    IER ...":GOTO 740                      <168>
555 PRINT"{HOME,RED,DOWN}"SPC(13)"OYYYYYYY
    YYYYP":PRINT SPC(13)"H B A S A R N"    <226>
560 PRINT SPC(13)"LPPPPPPPPPPPPS"          <242>
565 IF XX=1 THEN 655                       <151>
570 PRINT SPC(9)"{BLACK,DOWN}EIN{3SPACE}BE
    FREUNDETER"                            <221>
575 PRINT SPC(9)"{DOWN}HAENDLER IST BEREIT
    ,":PRINT SPC(9)"{DOWN}IHNEN ZWEI DER F
    OLGEN-"                                <227>
580 PRINT SPC(9)"{DOWN}DEN GEGENSTAENDE ZU
    ":PRINT SPC(9)"{DOWN}UEBERLASSEN :"    <135>
585 PRINT SPC(9)"{DOWN,RVSON}1{RVOFF,SPACE
    }GLASKUGEL{SPACE,RVSON}2{RVOFF,SPACE}K
    AEFIG"                                 <253>
590 PRINT SPC(9)"{DOWN,RVSON}3{RVOFF,SPACE
    }AMULETT{3SPACE,RVSON}4{RVOFF,SPACE}BU
    CH"                                    <140>
595 PRINT"{4DOWN,2SPACE}BITTE ENTSPRECHEND
    E NUMMERN EINGEBEN"                    <095>
600 GET I$:IF I$=""THEN 600                <193>
605 IF ASC(I$)<49 OR ASC(I$)>52 THEN 600   <187>
610 ON VAL(I$)GOSUB 635,640,645,650        <059>
615 GET J$:IF J$=""OR J$=I$THEN 615        <068>
620 IF ASC(J$)<49 OR ASC(J$)>52 THEN 615   <236>
625 ON VAL(J$)GOSUB 635,640,645,650        <106>
630 XX=1:GG=GG+2:GOSUB 7000:GOTO 655       <110>
635 POKE 1633,49:G(1)=0:RETURN             <002>
640 POKE 1645,50:G(2)=0:RETURN             <183>
645 POKE 1713,51:G(3)=0:RETURN             <013>
650 POKE 1725,52:G(4)=0:RETURN             <213>
655 YY=0:FOR I=1 TO 4:IF G(I)=0 THEN YY=YY
    +1                                     <004>
657 NEXT:IF YY<>2 THEN 710                 <007>
660 PRINT"{HOME,BLACK,5DOWN}"SPC(9)"WOLLEN
     SIE DIE GEGEN-"                       <138>
665 PRINT SPC(9)"{DOWN}STAENDE, DIE SIE HI
    ER":PRINT SPC(9)"{DOWN}ERHALTEN HABEN,
     NOCH{2SPACE}"                         <183>
670 PRINT SPC(9)"{DOWN}EINMAL UMTAUSCHEN ?
    ":PRINT SPC(9)"{DOWN,8SPACE}(J/N)"     <072>
675 PRINT:PRINT SPC(8)CL$:PRINT:PRINT SPC(
    8)CL$:GOSUB 860                        <089>
680 GET I$:IF I$="N"THEN 710               <185>
685 IF I$<>"J"THEN 680                     <196>
690 FOR I=1 TO 4:G(I)=20:NEXT:GG=GG-2      <134>
695 PRINT"{HOME,4DOWN}":FOR I=1 TO 9:PRINT
     SPC(8)CL$:NEXT                        <170>
700 PRINT"{HOME,5DOWN}"SPC(12)"WAEHLEN SIE
     !":PRINT"{7DOWN}"                     <218>
705 GOTO 585                               <189>
710 PRINT"{HOME,4DOWN,BLACK}":FOR I=1 TO 9
    :PRINT SPC(8)CL$:NEXT                  <038>
715 PRINT"{HOME,3DOWN}":FOR I=1 TO 5:PRINT
     SPC(13)"T"SPC(11)"T":NEXT             <056>
720 PRINT SPC(13)"{CYAN}OYYYYYYYYYYYP":PRI
    NT SPC(13)"HOYP OYP OYPN"              <026>
725 PRINT SPC(13)"HH{RED}Q{CYAN}N H{RED}Q{
    CYAN}N H{RED}Q{CYAN}NN"                <112>
730 PRINT SPC(13)"HLPS LPS LPSN":PRINT SPC
    (13)"LPPPPPPPPPPPS"                    <190>
740 POKE 214,18:PRINT:PRINT SPC(9)"{BLACK}
    RICHTUNGEN :{SPACE,WHITE}"RI$(P):GOTO
    750                                    <140>
748 REM **** KOMMANDOAUSWERTUNG ****       <127>
749 FOR I=1 TO 2500:NEXT                   <034>
750 ZA=0:GOSUB 7000                        <194>
755 GOSUB 860:INPUT"{YELLOW,SPACE}KOMMANDO
     :";K$                                 <161>
760 IF LEFT$(K$,1)=" "THEN K$=RIGHT$(K$,LE
    N(K$)-1):GOTO 760                      <093>
765 IF LEN(K$)=1 THEN 870                  <193>
770 FOR I=1 TO LEN(K$):IF MID$(K$,I,1)=" "
     THEN 780                              <130>
772 NE                                     <020>
775 GOSUB 860:PRINT"{10SPACE}UNKORREKTE EI
    NGABE !":GOTO 749                      <239>
780 V$=LEFT$(K$,I-1):H$=RIGHT$(K$,LEN(K$)-
    I)                                     <064>
785 IF LEFT$(H$,1)=" "THEN H$=RIGHT$(H$,LE
    N(H$)-1):GOTO 785                      <085>
790 IF V$="SAGE"THEN 6500                  <143>
795 FOR I=1 TO 14:IF LEFT$(V$(I),LEN(V$))=
    V$THEN X=I:GOTO 805                    <056>
800 NEXT:NV$=V$:GOTO 815                   <090>
805 FOR I=1 TO 21:IF LEFT$(H$(I),LEN(H$))=
    H$THEN Y=I:GOTO 825                    <054>
810 NEXT:NV$=H$                            <171>
815 GOSUB 860:PRINT"{UP,SPACE}TUT MIR LEID
    , ABER ICH VERSTEHE"                   <153>
820 PRINT"{DOWN,SPACE}'"NV$"' NICHT.":GOTO
     749                                   <229>
825 IF X>9 THEN X=X-5                      <063>
830 FOR I=1 TO LEN(KB$(X)):IF MID$(KB$(X),
    I,1)=CHR$(Y+64)THEN 950                <129>
835 NEXT:GOSUB 860                         <168>
840 PRINT"{UP,SPACE}TUT MIR LEID, ABER ICH
     VERSTEHE IHREN"                       <121>
845 PRINT"{DOWN,SPACE}BEFEHL NICHT.":GOTO
    749                                    <052>
860 POKE 214,20:PRINT:FOR I=1 TO 3:PRINT E
    F$:NEXT:PRINT"{3UP}":RETURN            <242>
869 REM **** BEWEGEN ****                  <008>
870 FOR I=1 TO 6:IF K$=MID$("NSWOHR",I,1)T
    HEN 880                                <027>
875 NEXT:GOTO 775                          <049>
880 IF GS=1 OR ZB=1 OR MR=1 THEN 7600      <245>
882 FOR I=1 TO LEN(RI$(P))STEP 2           <172>
885 IF MID$(RI$(P),I,1)=K$THEN 900         <175>
890 NEXT:GOSUB 860                         <223>
895 PRINT"{3SPACE}IN DIESE RICHTUNG GEHT E
    S NICHT":GOTO 749                      <165>
900 SC=0:ZI=VAL(MID$(ZI$(P),I,2))          <086>
905 IF P=16 AND ZI=13 THEN 925             <246>
910 IF P=13 AND ZI=9 AND G(8)=0 AND V=1 TH
    EN 7660                                <096>
915 IF ZI=0 THEN 7680                      <102>
920 P=ZI:GOTO 180                          <245>
```

Listing Castle of Doom. (Fortsetzung)

```
925 IF G(8)=0 THEN ON V GOTO 930,7650,7670 <115>
930 GOSUB 860:PRINT"{UP,SPACE}HINTER IHNEN
    LOEST SICH EINE STEIN-"                <107>
935 PRINT"{DOWN,SPACE}LAWINE, DIE DEN WEG
    VERSCHUETTET.":RI$(16)="W-O":ZI$(16)="
    1517"                                  <130>
940 FOR I=1 TO 2500:NEXT:P=ZI:GOTO 180     <221>
949 REM ** SPRUNG ZUR BEF.AUSFUEHRUNG **   <127>
950 IF X=4 AND Y<13 THEN 7100              <207>
955 IF X=5 THEN 7200                       <088>
960 ZN=1000+(X-1)*500+(Y-1)*20:ZN$=STR$(ZN
    )                                      <106>
965 FOR I=1 TO 4:NZ=VAL(MID$(ZN$,I+1,1))   <003>
970 POKE(2060+I),(NZ+48)                   <252>
975 NEXT                                   <223>
980 GOSUB 860:GOTO 1                       <189>
998 REM *** BEFEHLSAUSFUEHRUNG ***         <180>
999 PRINT"{2SPACE}OK !":GOTO 749           <119>
1000 IF Y=1 OR Y=7 OR Y=16 OR Y=19 OR Y=20
     THEN E$="E ":GOTO 1002                <152>
1001 E$=" "                                <028>
1002 PRINT" HIER IST KEIN"E$H$(Y):GOTO 749 <188>
1020 IF G(Y)=0 OR G(Y)=P THEN 999          <253>
1022 GOTO 1000                             <200>
1060 GOTO 1020                             <048>
1180 IF G(10)<>0 AND G(10)<>P THEN 1000    <080>
1182 IF V<>2 THEN PRINT" DER "H$(10)" LAES
     ST SICH NICHT OEFFNEN":GOTO 749       <060>
1184 IF G(9)<>21 THEN 999                  <250>
1186 PRINT" IM "H$(10)" BEFINDET SICH EIN
     "H$(9)" !!":G(9)=P:GOTO 749           <202>
1300 IF P=18 AND RI$(18)="H"AND V=3 THEN 1
     306                                   <143>
1302 IF P=1 OR P>13 THEN 1000              <206>
1304 PRINT" HIER IST KEINE GESCHLOSSENE TU
     ER":GOTO 749                          <085>
1306 IF G(11)<>0 THEN PRINT" DIE TUER IST
     VERSCHLOSSEN !":GOTO 749              <214>
1308 RI$(18)="O-H":ZI$(18)="1909":RI$(19)=
     "S-W":ZI$(19)="1718":GOTO 180         <240>
1340 IF P<>11 THEN 1000                    <152>
1342 IF V=3 OR G(10)<>21 THEN 999          <118>
1344 IF G(11)<>0 THEN PRINT" DER SCHRANK I
     ST VERSCHLOSSEN":GOTO 749             <067>
1346 PRINT"{UP,3SPACE}IM SCHRANK BEFINDET
     SICH"                                 <138>
1348 PRINT"{DOWN,3SPACE}EIN GOLDENER BALL"
     :G(10)=11:GOTO 749                    <067>
1400 PRINT" DAS IST UNMOEGLICH !":GOTO 749 <168>
1580 IF G(5)<>P THEN 1000                  <120>
1582 IF P=18 THEN PRINT" ES IST ZU DUNKEL
     HIER !":GOTO 749                      <080>
1584 IF G(2)=0 THEN 1590                   <148>
1586 PRINT"{UP,SPACE}SIE BRAUCHEN EINEN BE
     HAELTER"                              <021>
1588 PRINT"{DOWN,SPACE}UM DEN FROSCH ZU FA
     NGEN !":GOTO 749                      <016>
1590 IF GG=5 THEN PRINT" SIE TRAGEN SCHON
     5 GEGENSTAENDE !":GOTO 749            <245>
1592 G(5)=0:GG=GG+1:PRINT" OK !":GOTO 749  <114>
1740 IF MR=1 THEN 7640                     <254>
1742 IF V=3 AND P=9 AND MO=0 THEN PRINT" V
     ON HIER AUS GEHT DAS NICHT !":GOTO 74
     9                                     <028>
1744 GOTO 1000                             <160>
1780 IF GS<>1 THEN 1000                    <100>
1782 PRINT"{2SPACE}DER GEIST MAG DAS NICHT
      !!":FOR I=1 TO 2000:NEXT:GOTO 7630   <134>
2080 IF G(5)=P THEN PRINT" DER FROSCH LAES
     ST SICH NICHT KUESSEN":GOTO 749       <046>
2082 IF G(5)<>0 THEN 1000                  <050>
2084 IF V<>1 OR P<>6 THEN 999              <226>
2086 PRINT" ES KNALLT UND STINKT...":FOR I
     =1 TO 2000:NEXT:GOTO 7800             <107>
2240 IF MR<>1 THEN 1000                    <084>
2242 PRINT"{UP,SPACE}DAS MONSTER SCHLAEGT
     WILD UM SICH"                         <046>
2244 PRINT"{DOWN,SPACE}UND SCHLEUDERT SIE
     AUS DEM BRUNNEN":MR=0                 <141>
2246 FOR I=1 TO 3000:NEXT:P=9:GOTO 180     <027>
2260 IF ZB<>1 THEN 1000                    <207>
2262 FOR I=0 TO 15:POKE 53281,I:POKE 53280
     ,I:FOR J=1 TO 100:NEXT:NEXT:ZB=0      <117>
2264 POKE 53280,5:POKE 53281,5:ZK=ZK+1:IF
```

```
     ZK=3 THEN 7620                        <217>
2266 P=2+INT(13*RND(TI)):GOTO 180          <166>
2280 IF GS<>1 THEN 1000                    <092>
2282 IF TI$<"001000"THEN 7630              <008>
2284 GE=1:GS=0:FOR I=832 TO 892 STEP 3:POK
     E I,0:POKE I+1,0:POKE I+2,0:NEXT      <091>
2286 PRINT"{UP,2SPACE}DER GEIST HAT IHREN
     MUNDGERUCH":PRINT"{DOWN,2SPACE}NICHT
     VERTRAGEN !"                          <179>
2288 GOTO 749                              <073>
2400 IF P<>10 OR FA=1 THEN 1000            <200>
2402 IF V<>1 THEN 999                      <115>
2404 IF V=1 THEN FA=1:RI$(10)="N-W-R":ZI$(
     10)="070919"                          <128>
2406 RI$(19)="S-H":ZI$(19)="1710":GOTO 180 <231>
2820 IF P<>3 OR LEN(RI$(3))=5 THEN 1000    <136>
2822 PRINT" DAS GEHT NICHT !":GOTO 749     <038>
2840 IF P<>11 THEN 1000                    <128>
2842 PRINT" DER SCHRANK IST ZU SCHWER !":G
     OTO 749                               <253>
2900 IF P<>10 OR FA=1 THEN 1000            <192>
2902 GOTO 2822                             <120>
3560 IF GS=1 THEN 7630                     <004>
3562 IF G(4)<>0 THEN PRINT" ICH HABE DAS B
     UCH NICHT !":GOTO 749                 <053>
3564 IF V=3 THEN 3568                      <122>
3566 PRINT" ZAUBERWORT : "Z1$:GOTO 749     <020>
3568 PRINT"{UP,SPACE}ZAUBERWOERTER : "Z1$:
     PRINT"{DOWN}"SPC(17)Z2$:GOTO 749      <068>
3660 IF G(9)<>0 THEN PRINT" ICH HABE KEINE
     N ZETTEL":GOTO 749                    <193>
3662 PRINT" ZAUBERWORT: "Z2$:FOR I=1 TO 25
     00:NEXT                               <029>
3664 PRINT"{UP,SPACE}DER ZETTEL VERSCHWIND
     ET !!!":GG=GG-1:G(9)=21:GOTO 749      <137>
3860 IF P<>12 OR SC=0 THEN 1000            <155>
3862 PRINT"{UP,2SPACE}TUT MIR LEID, ABER E
     S SCHEINT"                            <140>
3864 PRINT"{DOWN,2SPACE}EINE FREMDE SPRACH
     E ZU SEIN":GOTO 749                   <120>
4300 IF P<12 OR(V=3 AND P=18)THEN PRINT" O
     K...NICHTS PASSIERT !":GOTO 749       <227>
4302 GOTO 1000                             <178>
4320 IF P<>3 OR LEN(RI$(3))=5 THEN 1000    <110>
4322 PRINT"{UP,SPACE}HINTER DEM "H$(17)    <137>
4324 PRINT"{DOWN,SPACE}SCHEINT EIN HOHLRAU
     M ZU SEIN !":GOTO 749                 <071>
4380 IF V=2 AND P=12 AND LEN(RI$(12))=3 TH
     EN 4388                               <088>
4382 IF P=19 AND G(8)=21 THEN 4392         <197>
4384 IF P>14 AND P<18 THEN 1000            <017>
4386 PRINT" OK...NICHTS BESONDERES FESTZUS
     TELLEN !":GOTO 749                    <226>
4388 POKE 214,11:PRINT:PRINT SPC(18)"{YELL
     OW,RVSON}U@/H":GOSUB 860              <023>
4390 PRINT" EINE SCHRIFT ERSCHEINT...":SC=
     1:GOTO 749                            <008>
4392 G(8)=19:PRINT" AN DER WAND HAENGT EIN
      SCHWERT !":GOTO 749                  <098>
4400 IF P=10 AND FA=0 THEN PRINT" KLINGT H
     OHL...":GOTO 749                      <150>
4402 GOTO 1000                             <024>
4580 IF G(5)<>0 AND G(5)<>P THEN 1000      <154>
4582 PRINT" DER FROSCH ENTWISCHT IHNEN !"  <114>
4584 IF G(5)<>18 OR V<>1 THEN 4588         <059>
4586 GOTO 749                              <085>
4588 IF G(5)=0 THEN GG=GG-1                <001>
4590 G(5)=1+INT(19*RND(1)):GOTO 749        <177>
4740 IF MR=0 THEN 1000                     <108>
4742 PRINT" GUTE IDEE, ABER DAS GEHT NICHT
      !":GOTO 749                          <192>
4760 IF ZB=0 THEN 1000                     <231>
4762 GOSUB 4770:IF Y=0 OR G(Y)<>0 THEN 749 <040>
4764 IF Y=8 AND ZA=2 THEN 7820             <203>
4766 GOTO 7620                             <086>
4770 INPUT" WOMIT ";WM$                    <216>
4771 FOR I=1 TO LEN(WM$):IF MID$(WM$,I,1)=
     " "THEN 4773                          <058>
4772 NEXT:GOTO 4774                        <093>
4773 WM$=RIGHT$(WM$,LEN(WM$)-I)            <076>
4774 FOR Y=1 TO 12:IF WM$=LEFT$(H$(Y),LEN(
     WM$))THEN 4776                        <052>
4775 NEXT:GOSUB 860:PRINT" ICH VERSTEHE '"
     WM$"' NICHT !":Y=0:RETURN             <132>
```

Listing Castle of Doom. (Fortsetzung)

```
4776 IF G(Y)<>0 THEN GOSUB 860:PRINT" ICH
     HABE DIESEN GEGENSTAND NICHT !"          <240>
4777 RETURN                                   <007>
4780 IF GS=0 THEN 1000                        <116>
4782 GOSUB 4770:IF Y=0 OR G(Y)<>0 THEN 749 <060>
4784 GOSUB 860:ON V GOTO 4785,4790,4794       <037>
4785 IF G(8)=0 THEN PRINT" SIE SIND GESTOL
     PERT, UND...":FOR I=1 TO 2000:NEXT:GO
     TO 7630                                  <131>
4786 IF ZA=1 AND Y=7 THEN 4798                <205>
4788 GOTO 7630                                <140>
4790 IF ZA=1 AND Y=3 THEN 4798                <193>
4792 GOTO 7630                                <144>
4794 IF Y=1 THEN PRINT" DER GEIST IST TOT,
      ABER...":FOR I=1 TO 2000:NEXT           <239>
4796 GOTO 7630                                <148>
4798 POKE VI+21,0:GS=0:GE=1:PRINT" OK...DE
     R GEIST IST TOT !":GOTO 749              <000>
5000 IF G(1)<>0 AND G(1)<>P THEN 1000         <066>
5002 PRINT" WOZU SOLLTE DAS GUT SEIN ?":GO
     TO 749                                   <019>
5020 IF G(2)<>0 AND G(2)<>P THEN 1000         <150>
5022 PRINT"{UP,SPACE}SO GEHT MAN NICHT MIT
      FREMDEM"                                <112>
5024 PRINT"{DOWN,SPACE}EIGENTUM UM !":GOTO
      749                                     <007>
5300 IF P=18 AND RI$(18)="H"AND V=3 THEN 5
     306                                      <081>
5302 IF P=1 OR P>13 THEN 1000                 <140>
5304 PRINT"WIESO, HIER IST KEINE VERSCHLOS
     SENE TUER":GOTO 749                      <174>
5306 PRINT" DIE TUER IST ZU STABIL...":GOT
     O 749                                    <112>
5320 GOSUB 4770:IF Y=0 OR G(Y)<>0 THEN 749 <090>
5322 IF(V=1 AND Y=8)OR(V=3 AND ZA=2 AND Y=
     8)THEN 7247                              <149>
5324 GOSUB 860:PRINT" DAS GEHT NICHT !":GO
     TO 749                                   <028>
5340 IF P<>11 THEN 1000                       <086>
5342 PRINT"{UP,SPACE}DER SCHRANK LAESST SI
     CH NICHT"                                <006>
5344 PRINT"{DOWN,SPACE}MIT GEWALT OEFFNEN
     !":GOTO 749                              <034>
5380 IF P>14 AND P<18 THEN 1000               <051>
5382 PRINT"SIE KOENNEN DIE WAND NICHT ZERS
     TOEREN !":GOTO 749                       <036>
6499 REM *** BEFEHL 'SAGE' ***                <141>
6500 GOSUB 860:IF H$=Z1$THEN ZA=1:GOTO 650
     6                                        <086>
6502 IF H$=Z2$THEN ZA=2:GOTO 6506             <009>
6504 GOTO 7690                                <015>
6506 ON V GOTO 6508,6512,6520                 <064>
6508 IF ZA=1 AND G(7)=0 AND GS=1 AND G(4)=
     0 THEN PRINT" DIE KEULE VIBRIERT...":
     GOTO 6528                                <059>
6510 GOTO 6526                                <157>
6512 IF ZA=1 AND P=12 AND SC=1 THEN RI$(12
     )="N-W-O":ZI$(12)="080013":GOTO 415      <213>
6514 IF ZA=1 AND G(3)=0 AND GS=1 THEN POKE
      VI+39,0:GOTO 755                        <133>
6516 IF ZA=2 AND G(8)=0 AND ZB=1 AND G(4)=
     0 THEN PRINT" DAS SCHWERT LEUCHTET...
     ":GOTO 6528                              <096>
6518 GOTO 6526                                <165>
6520 IF ZA=1 AND G(3)=0 AND MO=0 AND P=9 T
     HEN PRINT" DAS AMULETT LEUCHTET...":G
     OTO 6528                                 <113>
6522 IF ZA<>2 OR G(4)<>0 OR P<>3 OR RI$(3)
     <>"W-O"THEN 6526                         <095>
6524 PRINT" DER SPIEGEL LEUCHTET...":GOTO
     6528                                     <134>
6526 PRINT" OK...ES TUT SICH NICHTS !":GOT
     O 749                                    <085>
6528 FOR I=1 TO 2000:NEXT:GOTO 755            <193>
6999 REM * BESITZ/GEGENSTD ANZEIGEN *         <167>
7000 PRINT"{HOME}";:FOR I=1 TO 20:PRINT"{8
     SPACE}":NEXT                             <226>
7002 PRINT"{HOME,BLACK,DOWN}SIE":PRINT"{DO
     WN}BESITZEN":SP=0                        <001>
7005 FOR I=1 TO 12:IF G(I)=0 THEN GOSUB 70
     50                                       <116>
7010 NEXT                                     <162>
7020 PRINT"{HOME}";:FOR I=1 TO 20:PRINT SP
     C(32)"{7SPACE}":NEXT                     <216>
7022 J=0:PRINT"{HOME,BLACK,DOWN}"SPC(32)"E
     S GIBT":PRINT SPC(32)"{DOWN}HIER :":S
     P=32                                     <122>
7025 FOR I=1 TO 12:IF G(I)=P THEN GOSUB 70
     50:J=J+1:IF J=6 THEN RETURN              <131>
7030 NEXT:RETURN                              <111>
7050 IF I=1 THEN PRINT SPC(SP)"{DOWN}GLAS-
     ":PRINT SPC(SP)"KUGEL":RETURN            <043>
7055 IF I=11 THEN PRINT SPC(SP)"{DOWN}SCHL
     UES":PRINT SPC(SP)"SEL":RETURN           <035>
7060 IF I=12 THEN PRINT SPC(SP)"{DOWN}KETT
     EN-":PRINT SPC(SP)"HEMD":RETURN          <081>
7065 PRINT SPC(SP)"{DOWN}"G$(I):RETURN        <055>
7099 REM *** BEFEHL 'NIMM' ***                <085>
7100 IF G(Y)=P THEN 7110                      <199>
7105 GOSUB 860:PRINT"{2SPACE}DIESER GEGENS
     TAND IST NICHT HIER !":GOTO 749          <197>
7110 IF GG<5 THEN 7120                        <080>
7115 GOSUB 860:PRINT"{2SPACE}SIE HABEN BER
     EITS 5 GEGENSTAENDE !":GOTO 749          <100>
7120 IF Y=5 THEN GOSUB 860:PRINT"{4SPACE}D
     ER FROSCH HUEPFT IHNEN DAVON !":GOTO
     749                                      <252>
7125 G(Y)=0:GG=GG+1:GOSUB 860:PRINT"{3SPAC
     E}OK !":FOR I=1 TO 1000:NEXT:GOTO 750 <010>
7199 REM *** BEFEHL 'WIRF' ***                <088>
7200 IF G(Y)<>0 THEN GOSUB 860:PRINT" ICH
     HABE DIESEN GEGENSTAND NICHT !":GOTO
     749                                      <209>
7205 GOSUB 860:INPUT"{2SPACE}WOHIN ";WH$      <085>
7210 FOR I=1 TO LEN(WH$):IF MID$(WH$,I,1)=
     " "THEN 7220                             <071>
7215 NEXT:GOTO 7225                           <154>
7220 WH$=RIGHT$(WH$,LEN(WH$)-I)               <059>
7225 IF WH$="BODEN"OR WH$="WEG"OR WH$="EGA
     L"OR WH$="WAND"THEN 7235                 <042>
7230 ON V GOTO 7240,7270,7330                 <255>
7235 GOSUB 860:PRINT"{2SPACE}OK !":G(Y)=P:
     GG=GG-1:GOTO 749                         <037>
7240 IF Y<>8 OR WH$<>"SPIEGEL"OR P<>3 OR L
     EN(RI$(3))=5 THEN 7255                   <189>
7245 G(Y)=3:GG=GG-1                           <058>
7247 RI$(3)="N-W-O":ZI$(3)="010204":GOSUB
     860                                      <013>
7250 PRINT"{2SPACE}DER SPIEGEL ZERBRICHT !
     ":FOR I=1 TO 2000:NEXT:GOTO 180          <086>
7255 IF Y<>10 OR WH$<>"BRUNNEN"OR P<>9 OR
     G(5)<>18 THEN 7380                       <235>
7260 GOSUB 860:PRINT"{UP,2SPACE}AUS DEM BR
     UNNEN KOMMT EIN FROSCH"                  <121>
7265 PRINT"{DOWN,2SPACE}MIT DEM BALL IM MU
     ND.":G(10)=9:G(5)=9:GG=GG-1:GOTO 749     <041>
7270 IF Y<>3 OR WH$<>"GEIST"OR GS<>1 THEN
     7290                                     <194>
7275 IF ZA<>1 THEN 7630                       <186>
7280 POKE VI+21,0:GOSUB 860:PRINT"{4SPACE}
     SIE HABEN DEN GEIST BESIEGT !"           <212>
7285 GE=1:GS=0:G(Y)=7:GG=GG-1:GOTO 749        <042>
7290 IF Y<>6 OR P<>10 OR WH$<>"QUADRAT"OR
     FA=1 THEN 7305                           <251>
7295 FA=1:G(6)=10:GG=GG-1:RI$(10)="N-W-R":
     ZI$(10)="070919"                         <198>
7300 RI$(19)="S-H":ZI$(19)="1710":GOTO 180 <043>
7305 IF Y<>4 OR P<>3 OR WH$<>"SPIEGEL"OR L
     EN(RI$(3))=5 THEN 7315                   <022>
7310 GOTO 7245                                <155>
7315 IF Y<>8 OR ZB<>1 OR WH$<>"ZAUBERER"TH
     EN 7380                                  <062>
7320 IF ZA<>2 THEN 7620                       <096>
7325 POKE VI+21,0:FOR I=1 TO 2000:NEXT:GOT
     O 7820                                   <002>
7330 IF Y<>1 OR WH$<>"GEIST"OR GS<>1 THEN
     7340                                     <237>
7335 GOTO 7280                                <244>
7340 IF Y<>3 OR WH$<>"BRUNNEN"OR P<>9 THEN
      7360                                    <034>
7345 G(3)=18:GG=GG-1:GOSUB 860:IF MO=1 THE
     N PRINT"{2SPACE}OK !":GOTO 749           <030>
7350 IF ZA=1 THEN PRINT" AUS DEM BRUNNEN E
     RTOENT EIN SCHREI...":MO=1:GOTO 749      <163>
```

Listing Castle of Doom. (Fortsetzung)

```
7355 G(3)=9:PRINT" DAS AMULETT KOMMT ZURUE
     CKGEFLOGEN !":GOTO 749                <096>
7360 IF Y<>8 OR WH$<>"SPIEGEL"OR P<>3 OR L
     EN(RI$(3))=5 THEN 7380                <027>
7365 IF ZA=2 THEN 7245                     <218>
7370 GOTO 7235                             <183>
7380 IF WH$="BRUNNEN"AND P=9 THEN G(Y)=18:
     GOTO 7415                             <249>
7382 IF WH$="QUADRAT"AND P=10 AND FA=0 THE
     N G(Y)=10:GOTO 7415                   <249>
7385 IF WH$="HOEHLE"AND P=17 THEN G(Y)=19:
     GOTO 7415                             <129>
7390 IF WH$="GEIST"AND GS=1 THEN 7630      <064>
7395 IF WH$="ZAUBERER"AND ZB=1 THEN 7620   <078>
7400 IF WH$="SPIEGEL"AND P=3 AND LEN(RI$(3
     ))=3 THEN 7235                        <053>
7405 GOSUB 860:PRINT" ICH VERSTEHE SIE NIC
     HT..."                                <000>
7410 FOR I=1 TO 2000:NEXT:GOTO 7205        <000>
7415 GG=GG-1:GOSUB 860:PRINT"{2SPACE}OK !"
     :GOTO 749                             <160>
7499 REM *** ERKLAERUNG ***                <178>
7500 PRINT"{CLR,DOWN,SPACE}SIE MUESSEN VER
     SUCHEN, EINE AUFGABE ZU"              <052>
7505 PRINT"{DOWN,SPACE}LOESEN, INDEM SIE D
     EM COMPUTER SAGEN,"                   <085>
7510 PRINT"{DOWN,SPACE}WAS ER TUN SOLL."   <086>
7515 PRINT"{DOWN,SPACE}ZUR{2SPACE}FORTBEWE
     GUNG{2SPACE}GEBEN{2SPACE}SIE NUR"     <177>
7520 PRINT"{DOWN,SPACE}ABKUERZUNGEN EIN (N
     -S-W-O-H-R = NORD -":REM N-S-W-O-H-R
     = NORD -                              <106>
7525 PRINT"{DOWN,SPACE}SUED - WEST - OST -
      HOCH - RUNTER)"                      <205>
7526 REM SUED - WEST - OST - HOCH - RUNTER <189>
7530 PRINT"{DOWN,SPACE}ANSONSTEN MUESSEN D
     IE BEFEHLE AUS ZWEI"                  <006>
7535 PRINT"{DOWN,SPACE}WOERTERN BESTEHEN (
     VERB/HAUPTWORT)"                      <090>
7540 PRINT"{DOWN,SPACE}DER COMPUTER VERSTE
     HT DIE VERBEN :"                      <096>
7545 PRINT"{DOWN,SPACE}OEFFNE,FANGE,KUESSE
     ,NIMM,LIES,WIRF,"                     <189>
7550 PRINT"{DOWN,SPACE}BERUEHRE/KLOPFE,ERS
     CHLAGE/TOETE,SAGE"                    <185>
7555 PRINT"{DOWN,SPACE}ZERBRECHE/ZERSTOERE
     {3SPACE}-{2SPACE}FERTIG ???"          <113>
7560 POKE 198,0:WAIT 198,1:GOTO 100        <239>
7599 REM **** VERLOREN ****                <067>
7600 IF ZB=1 THEN 7620                     <111>
7605 IF GS=1 THEN 7630                     <241>
7610 IF MR=1 THEN 7640                     <026>
7620 PRINT"{CLR,BLACK,DOWN,SPACE}DER ZAUBE
     RER HAT SIE IN EINE KROETE"           <145>
7625 PRINT"{DOWN,SPACE}VERWANDELT.":GOTO 7
     700                                   <006>
7630 PRINT"{CLR,BLACK,DOWN,5SPACE}DER GEIS
     T HAT SIE GEBISSEN.":GOTO 7700        <208>
7640 PRINT"{CLR,BLACK,DOWN,SPACE}DAS MONST
     ER HAT BESCHLOSSEN, SIE"              <231>
7645 PRINT"{DOWN,SPACE}ZU HEIRATEN.":GOTO
     7700                                  <030>
7650 PRINT"{CLR,BLACK,DOWN,SPACE}SIE SIND
     UNTERWEGS GESTOLPERT UND"             <042>
7655 PRINT"{DOWN,SPACE}HABEN SICH DABEI DE
     N KLEINEN FINGER":PRINT"{DOWN,SPACE}V
     ERSTAUCHT.":GOTO 7700                 <204>
7660 PRINT"{CLR,BLACK,DOWN,SPACE}AUS DEM T
     ORBOGEN HAT SICH EIN STEIN"           <110>
7665 PRINT"{DOWN,SPACE}GELOEST UND IST IHN
     EN AUF DEN FUSS":PRINT"{DOWN,SPACE}GE
     FALLEN.":GOTO 7700                    <149>
7670 PRINT"{CLR,BLACK,DOWN,SPACE}SIE SIND
     IN EINEN ERDRUTSCH GERATEN"           <184>
7675 PRINT"{DOWN,SPACE}UND HABEN SICH IHRE
      NEUE HOSE DRECKIG":PRINT"{DOWN,SPACE
     }GEMACHT.":GOTO 7700                  <206>
7680 PRINT"{CLR,BLACK,DOWN,SPACE}SIE SIND
     VOM BERG GEFALLEN UND HABEN"          <225>
7685 PRINT"{DOWN,SPACE}DABEI IHRE BRILLE Z
     ERBROCHEN.":GOTO 7700                 <167>
7690 PRINT"{CLR,BLACK,DOWN,8SPACE}SIE HABE
     N SICH IN EINE"                       <195>
7695 PRINT"{DOWN,10SPACE}BANANE VERWANDELT
      !"                                   <135>
7700 PRINT"{DOWN,SPACE}DESHALB MUESSEN SIE
      LEIDER AUFGEBEN."                    <096>
7705 PRINT"{3DOWN,YELLOW,SPACE}MOECHTEN SI
     E ES NOCH EINMAL PROBIEREN ?"         <068>
7710 GET I$:IF I$="J"THEN RUN 5            <238>
7715 IF I$<>"N"THEN 7710                   <154>
7720 END                                   <102>
7799 REM **** GEWONNEN ****                <082>
7800 PRINT"{CLR,YELLOW,DOWN,8SPACE}... UND
     {2SPACE}DER{2SPACE}"H$(5)" ...":FOR I
     =1 TO 1500:NEXT                       <112>
7805 PRINT"{UP,9SPACE}... VERWANDELT{2SPAC
     E}SICH ...":FOR I=1 TO 1500:NEXT      <074>
7810 PRINT"{UP}... IN EINE WUNDERSCHOENE P
     RINZESSIN ...":FOR I=1 TO 2000:NEXT   <236>
7815 PRINT"{CLR,5DOWN}":GOTO 7840          <195>
7820 POKE VI+21,0:PRINT"{CLR,YELLOW,6DOWN,
     4SPACE}SIE HABEN DEN ZAUBERER BESIEGT
      !!!":GOTO 7840                        <104>
7830 PRINT"{CLR,YELLOW,6DOWN,4SPACE}SIE HA
     BEN DEN SCHATZ GEFUNDEN !!!"          <016>
7840 PRINT SPC(13)"{2DOWN}GRATULIERE,":PRI
     NT"{DOWN,5SPACE}SIE HABEN IHR ZIEL ER
     REICHT !!"                            <142>
7850 PRINT"{3DOWN,6SPACE}WOLLEN SIE WEITER
      SPIELEN ?"                           <198>
7860 GET I$:IF I$="J"THEN RUN 5            <132>
7870 IF I$<>"N"THEN 7860                   <229>
7880 END                                   <006>
8000 DATA OEFFNE,BDJPRU,FANGE,EMO,KUESSE,E
     MNOU,NIMM,ABCDEFGHIJKLQRU             <201>
8005 DATA WIRF,ABCDEFGHIJKL,LIES,DIS,BERUE
     HRE,PQTU,TOETE,EMNO                   <006>
8010 DATA ZERSTOERE,ABPQRT,WERFE,LESE,KLOP
     FE,ERSCHLAGE,ZERBRECHE                <166>
8015 DATA GLASKUGEL,KAEFIG,AMULETT,BUCH,FR
     OSCH,HELM,KEULE,SCHWERT,ZETTEL,BALL   <092>
8020 DATA SCHLUESSEL,KETTENHEMD,MONSTER,ZA
     UBERER,GEIST,TUER,SPIEGEL,SCHRANK     <042>
8025 DATA SCHRIFT,WAND,QUADRAT             <148>
8030 DATA S,03,S-O,0503,W-O,0204,W-H,0311,
     N-S,0208,O,07,S-W,1006,N-O,0509       <063>
8035 DATA S-W-O-R,13081018,N-W,0709,R,04,W
     -O,0013,N-W-O,091214,W-O,1300,O,16    <138>
8040 DATA W-O-H,151713,N-W,1916,H,09,S,17  <147>
8045 DATA 20,20,20,20,18,2,21,21,21,21,6,5 <191>
8050 DATA SILBER,BLUME,GOLD,RING,KUPFER,HO
     RN,PLATIN,STERN                       <026>
8054 REM SPRITE 1 - GEIST                  <103>
8055 DATA 0,127,0,0,255,192,1,206,96,3,255
     ,240,1,255,240,1,245,248,1,255,248    <204>
8060 DATA 3,255,248,7,255,248,15,255,252,3
     1,255,254,3,255,240,3,255,224         <173>
8065 DATA 3,255,224,7,255,224,7,255,224,7,
     255,240,15,255,240,15,255,248         <105>
8070 DATA 31,255,252,59,187,110            <043>
8074 REM SPRITE 2 - MONSTER                <214>
8075 DATA 12,0,48,30,0,120,3,0,192,1,129,1
     28,3,195,192,15,231,240,62,126,124    <031>
8080 DATA 124,60,62,248,60,31,254,126,127,
     255,255,255,127,255,254,127,255,254   <034>
8085 DATA 63,90,252,30,165,120,31,255,248,
     7,255,224,4,126,32,4,60,32            <195>
8090 DATA 31,0,248,42,129,84               <152>
8094 REM SPRITE 3 - ZAUBERER               <197>
8095 DATA 0,15,224,0,63,128,0,127,128,0,21
     9,0,1,255,0,3,255,128,31,255,240      <175>
8100 DATA 127,255,252,127,255,252,111,255,
     236,111,255,236,103,255,204,98,170,14
     0                                     <252>
8105 DATA 99,255,140,71,255,196,7,199,192,
     15,131,224,15,1,224,14,0,224          <202>
8110 DATA 28,0,112,60,0,120                <212>
8200 REM                                   <039>
8201 REM **********************            <161>
8202 REM *   CASTLE OF DOOM   *            <007>
8203 REM * VON B. WEISSBECKER *            <102>
8204 REM *  MAERZ/APRIL 1984  *            <139>
8205 REM **********************            <165>
```


Listing Castle of Doom. (Schluß)

APOCALYPSE NOW

Im Universum versteckt sich die interplanetarische Spinne Plexar. Sie können eine Katastrophe abwenden, wenn Sie den Weltraum von dem schrecklichen Ungeheuer befreien.

Bevor Sie mit dem apokalyptischen Kampf gegen die interplanetarische Spinne beginnen, müssen Sie auftanken. Doch nach 30 Energieeinheiten beginnt ein feindlicher Hubschrauber auf Sie zu schießen. Ist dieses Hindernis hinter Ihnen, geht es links weiter. Im oberen Eck erscheint ein Tunnel, den Sie durchfliegen. Dabei müssen Sie eine Lasersperre überwinden oder einen Mann retten. Dieser sollte bei der Aktion möglichst weit links stehen. Haben Sie diese Hürde überwunden, schließt sich hinter Ihnen eine Mauer. Durch diese müssen Sie sich links unten durchschießen. Auf dem nächsten Bild gilt es, eine Barrikade zu durchschießen, die aber mit Vorsicht zu genießen ist. Die nachfolgenden Panzer können Sie nur abschießen, wenn Sie beide zugleich treffen.

Dann kommen Sie zu der Spinne Plexar. Achten Sie vor allem auf die Lasersperren. Wenn Sie nun die Spinne vernichtet haben, fliegen Sie zurück, aber achten Sie darauf, daß Sie möglichst hoch in den nächsten Bildschirm fliegen.

Zum guten Schluß können Sie auf dem Energielager landen und sich von den Strapazen erholen, oder Sie spielen weiter. Das hat aber nur Sinn, wenn Sie noch zwei Leben haben.

(Helmut Burgemeister/Helmut Bölcskei/tr)

Die wichtigsten Variablen:

X und Y für Hubschrauber
X1 bis X7 und Y1 bis Y7 für andere Sprites
HU = feindliche Hubschrauberzahl
LD = Plexar abgeschossen?
A = Hubschrauberstellung
E = Schußvariable

```
1 REM +++++++++++++++++++                        <066>
2 REM + APOCALYPSE NOW +                         <175>
3 REM +++++++++++++++++++**********              <065>
4 REM * BY HELMUT BURGEMEISTER *                 <025>
5 REM * AND HELMUT BOELCSKEI   *                 <228>
6 REM ****************************               <046>
7 REM ↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑               <216>
8 REM ↑   MASCHINEN PROGRAMM     ↑               <008>
9 REM ↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑               <218>
10 PRINT"{CLR,6DOWN}BITTE WARTEN !"              <064>
20 POKE 53281,0                                  <198>
30 POKE 53280,0                                  <080>
40 FOR I=832 TO 832+33:READ A:POKE I,A:NEX
   T                                             <174>
50 DATA 120,169,51,133,1,169,0,133,95,133,
   90,133,88,169,208,133,96,169,240              <104>
60 DATA 133,89,169,224,133,91,32,191,163,1
   69,55,133,1,88,96                             <133>
80 SYS 832:POKE 850,160:SYS 832                  <141>
100 REM ←←←←←←←←←←←←←←←←←←←←←←←←←←←←             <106>
101 REM ← DATEN FUER NEUE ZEICHEN ←              <110>
102 REM ←←←←←←←←←←←←←←←←←←←←←←←←←←←←             <108>
130 POKE 56576,PEEK(56576)AND 252                <129>
135 POKE 53272,8:POKE 648,192                    <165>
140 PRINT"{CLR,6DOWN}BITTE WARTEN !"             <194>
150 FOR I=280 TO 559:READ A:POKE 57344+I,A
    :NEXT                                        <095>
210 DATA 255,255,255,255,255,255,255,255         <036>
220 DATA 0,0,0,37,205,255,255,255                <030>
230 DATA 0,0,0,0,0,255,255,255                   <008>
240 DATA 137,66,36,24,153,36,66,137              <065>
245 DATA 255,129,129,255,129,129,255,255         <225>
250 DATA 3,13,49,193,193,49,13,3                 <101>
255 DATA 0,96,252,255,252,96,0,0                 <216>
260 DATA 60,60,60,24,24,0,0,0                    <185>
265 DATA 255,126,255,126,255,126,255,126         <085>
270 DATA 0,0,0,255,255,255,0,0                   <126>
275 DATA 24,60,24,63,88,88,36,66                 <112>
280 DATA 25,61,25,62,88,88,36,66                 <233>
285 DATA 0,0,60,32,60,32,32,32                   <152>
290 DATA 0,0,68,68,68,68,68,124                  <030>
295 DATA 0,0,124,64,124,64,64,124                <096>
300 DATA 0,0,32,32,32,32,32,60                   <007>
305 DATA 126,255,255,231,231,255,255,126         <065>
310 DATA 0,0,0,0,8,5,3,7                         <085>
315 DATA 0,0,0,0,12,62,127,255                   <200>
320 DATA 0,0,6,31,127,255,255,255                <031>
325 DATA 0,0,0,192,231,239,255,255               <081>
330 DATA 0,0,56,252,252,254,255,255              <079>
335 DATA 255,127,127,63,31,12,0,0                <070>
340 DATA 255,255,254,152,0,0,0,0                 <175>
345 DATA 255,191,31,15,3,1,0,0                   <174>
350 DATA 255,255,255,248,248,224,0,0             <255>
355 DATA 0,0,36,24,24,36,0,0                     <026>
360 DATA 65,39,18,207,63,191,127,255             <079>
365 DATA 160,68,88,225,250,252,254,255           <059>
370 DATA 255,255,255,255,255,255,255,255         <198>
375 DATA 255,255,243,195,195,192,0,0             <175>
380 DATA 0,0,35,254,35,0,0,0                     <042>
385 DATA 4,4,11,48,48,64,0,0                     <058>
390 DATA 0,0,196,127,196,0,0,0                   <205>
395 DATA 0,32,32,208,12,12,2,0                   <117>
400 REM +++++++++++++++++                        <033>
401 REM + SPRITES DATEN +                        <047>
402 REM +++++++++++++++++                        <035>
500 FOR I=50304 TO 50304+62:READ A:POKE I,
    A:NEXT                                       <101>
505 FOR I=50368 TO 50368+62:READ A:POKE I,
    A:NEXT                                       <251>
510 FOR I=50432 TO 50432+62:READ A:POKE I,
    A:NEXT                                       <206>
511 FOR I=50496 TO 50496+62:READ A:POKE I,
    A:NEXT                                       <097>
512 FOR I=50560 TO 50560+62:READ A:POKE I,
    A:NEXT                                       <049>
513 FOR I=50624 TO 50624+62:READ A:POKE I,
    A:NEXT                                       <216>
514 FOR I=50688 TO 50688+62:READ A:POKE I,
    A:NEXT                                       <107>
515 FOR I=50752 TO 50752+62:READ A:POKE I,
    A:NEXT                                       <057>
517 PRINT"{DOWN,3SPACE,BLUE,SPACE}APOCALYP
    SE NOW "                                     <151>
518 PRINT"{2DOWN,2SPACE,YELLOW}BY{SPACE,GR
    EEN}HELMUT{SPACE,CYAN}BURGEMEISTER{LIG
    .BLUE,DOWN,13LEFT,DOWN}AND HELMUT BOEL
    CSKEI{2SPACE}"                               <064>
519 DATA 0,0,0                                   <133>
520 DATA 0,0,0,255,255,255,0,48,0,3,255,19
    3,4,32,97,8,32,55,16,42,153                  <157>
525 DATA 96,34,1,128,79,131,128,66,6,128,1
    38,152,255,128,96,64,64,128                  <132>
530 DATA 32,33,0,16,33,0,143,254,0,129,2,0
    ,127,255,248,0,0,0,0,0                       <191>
540 DATA 0,0,0,0,0,0,31,255,240,0,40,0,0,1
    6,0,0,104,0,0,132,0                          <111>
545 DATA 1,2,0,6,249,128,4,252,128,11,254,
    64,9,252,64,4,0,128,3,3,0,0,252,0            <224>
550 DATA 1,50,0,2,1,0,4,0,128,28,0,224,0,0
    ,0,0,0,0                                     <198>
560 DATA 0,0,0,0,0,0,255,255,255,0,12,0,13
    1,255,112,135,4,32                           <239>
565 DATA 252,4,16,153,68,8,128,68,6,193,24
    2,1,96,66,1,24,65,1,6,1,255,1,2,2,0          <016>
570 DATA 132,4,0,132,8,0,127,241,0,64,129,
    31,255,254,0,0,0,0,0                         <163>
580 DATA 0,0,240,0,15,1,0,240,193,15,111,1
    03,240,240,65,3,32,145,4,42,3                <243>
```

Listing »APOCALYPSE NOW«.
Beachten Sie die Eingabehinweise auf Seite 16.

```
585 DATA 8,34,134,0,47,8,16,66,144,96,82,9
    6,128,128,128,128,129,0,143,65,0          <083>
590 DATA 240,33,0,64,46,8,32,242,240,31,15
    ,0,129,240,0,143,0,0,112,0,0              <044>
600 DATA 15,0,0,128,240,0,131,15,0,230,252
    ,240,156,240,15,137,4,192,192,84,32       <157>
605 DATA 97,68,16,16,244,0,9,66,8,6,82,6,1
    ,1,1,0,129,1,0,130,241,0,132,15           <125>
610 DATA 16,116,2,15,143,4,0,240,248,0,15,
    129,0,0,241,0,0,14                        <107>
620 DATA 0,0,0,0,0,0,0,3,0,0,15,0,0,63,0,0
    ,240,0,0,192,0,0,20,0,0,20,0,0,40,0       <095>
625 DATA 0,40,0,0,20,0,0,20,0,0,40,0,0,40,
    0,0,20,0,0,95,0,0,170,0,3,255,192         <016>
630 DATA 13,95,112,3,255,192                  <145>
640 DATA 0,0,0,0,0,0,0,192,0,0,240,0,0,63,
    0,0,15,0,0,3,0,0,20,0,0,20,0,0,40,0       <084>
645 DATA 0,40,0,0,20,0,0,20,0,0,40,0,0,40,
    0,0,20,0,0,95,0,0,170,0,3,255,192         <036>
650 DATA 13,95,112,3,255,192                  <165>
660 DATA 128,0,1,71,255,242,36,0,4,16,255,
    12,25,0,148,38,0,100,2,0,68,69,36,196     <027>
665 DATA 69,153,68,124,126,125,68,60,68,68
    ,90,68,68,129,68,67,0,196,66,255,68,68    <202>
670 DATA 20,36,40,34,20,16,65,24,47,255,22
    8,64,0,2,128,0,1                          <234>
700 REM ↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑         <087>
701 REM ↑ INETIALISIERUNG VARIABLEN ↑         <061>
702 REM ↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑         <089>
997 GOSUB 62130:POKE 50169,22:POKE 50168,1
    8                                         <009>
998 V=53248:POKE V+21,1:POKE V+42,14          <103>
999 X=250:Y=94:GOSUB 50000                    <033>
1000 X=250:Y=94:A=1:OT=1:V=53248:POKE 5016
     8,18:GOSUB 63000:O=PEEK(V+31):AB=AC      <140>
1001 POKE V+21,1:POKE V+29,3:POKE V+39,3:B
     =1:PRINT"{CLR}":Q=13:X2=0:Y2=90:B2=4:
     C2=3                                     <024>
1002 GOSUB 60100:POKE V+21,1                  <039>
1003 H=32:EN=10:PU=.05:B9=0:HU=0              <156>
1004 POKE V+2,X2:POKE V+3,Y2:HB=4             <048>
1005 POKE V+28,12:POKE V+37,2:POKE V+38,5     <221>
1006 POKE 50170,23:POKE 50171,24              <022>
1009 POKE SI+11,65:JS=PEEK(V+30):O=PEEK(V+
     31)                                      <001>
1010 REM HAUPTPROGRAMM                        <037>
1011 J=PEEK(56320):JS=PEEK(V+30)              <076>
1020 IF(J AND 4)=0 THEN X=X-5:A=A-1           <071>
1030 IF(J AND 8)=0 THEN X=X+5:A=A+1           <004>
1040 IF(J AND 1)=0 THEN Y=Y-5                 <084>
1050 IF(J AND 2)=0 THEN Y=Y+5                 <094>
1055 IF A>3 THEN A=3                          <129>
1056 IF A<1 THEN A=1                          <158>
1059 IF Y<85 THEN Y=85                        <204>
1060 ON A GOSUB 2000,2010,2020                <032>
1065 Y=Y+AB                                   <134>
1070 POKE V,X:POKE V+1,Y                      <209>
1080 IF(J AND 16)=0 THEN GOSUB 2050           <046>
1090 O=PEEK(V+31):IF O=1 THEN 3200            <127>
1098 IF X>250 THEN B=B-1:X=250:BI=0:HU=0:B
     9=0:POKE V+21,1:IF B<1 THEN B=1          <078>
1099 IF X<5 THEN B=B+1:X=5:BI=0               <006>
1100 ON B GOSUB 4500,5000,5500,6000           <061>
1110 POKE 49152+EN+40*3,H:EN=EN-PU:H=32       <251>
1888 IF EN>10 THEN 1010                       <153>
1889 GOSUB 61000:GOSUB 60900                  <230>
1890 IF EN<=0 THEN EN=0:AB=4:POKE SI+7,100
     :GOTO 1065                               <168>
1900 GOTO 1010                                <094>
1997 REM ↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑     <208>
1998 REM ↑BEWEGUNGSRICHTUNG VON SPRITE↑       <030>
1999 REM ↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑     <210>
2000 IF(J AND 4)=0 THEN POKE 50168,21:POKE
      SI+7,255:RETURN                         <200>
2001 POKE 50168,18:POKE SI+7,200              <092>
2009 RETURN                                   <033>
2010 POKE 50168,19                            <051>
2019 RETURN                                   <043>
2020 IF(J AND 8)=0 THEN POKE 50168,22:POKE
      SI+7,255:RETURN                         <229>
2021 POKE 50168,20:POKE SI+7,200              <082>
2029 RETURN                                   <053>
2049 REM ←←←←←←←←←←←←←←←←←←                   <050>
2050 REM ← SCHUSS ROUTINE ←                   <186>
2051 REM ←←←←←←←←←←←←←←←←←←                   <052>
2059 IF(J AND 4)=0 AND A=1 THEN GOSUB 3300
     :RETURN                                  <236>
2060 IF A=1 THEN GOSUB 3000                   <040>
2070 IF A=2 THEN GOSUB 3030                   <094>
2079 IF(J AND 8)=0 AND A=3 THEN GOSUB 3400
     :RETURN                                  <067>
2080 IF A=3 THEN GOSUB 3060                   <056>
2090 RETURN                                   <116>
3000 X1=INT(X/8-4):Y1=INT(Y/8-5)              <023>
3005 FOR E=X1 TO 0 STEP-1:GOSUB 61400         <094>
3006 PO=PEEK(49152+E+40*Y1):POKE 49152+E+4
     0*Y1,66:POKE 49152+E+40*Y1,PO            <031>
3007 IF PEEK(49152+E-1+40*Y1)<>32 THEN S=-
     1:GOTO 3100                              <007>
3010 NEXT E:RETURN                            <128>
3030 X1=INT(X/8):Y1=INT(Y/8-3)                <123>
3033 FOR E=Y1 TO 25 STEP 1:GOSUB 61400        <171>
3040 PO=PEEK(49152+X1+40*E):POKE 49152+X1+
     40*E,42:POKE 49152+X1+40*E,PO            <083>
3045 IF PEEK(49152+X1+40*E+40)<>32 THEN 31
     10                                       <063>
3050 NEXT E:RETURN                            <168>
3060 X1=INT(X/8+4):Y1=INT(Y/8-5)              <079>
3070 FOR E=X1 TO 39 STEP 1:GOSUB 61400        <185>
3075 PO=PEEK(49152+E+40*Y1):POKE 49152+E+4
     0*Y1,68:POKE 49152+E+40*Y1,PO            <231>
3077 IF PEEK(49152+E+1+40*Y1)<>32 THEN S=1
     :GOTO 3100                               <221>
3080 NEXT E:RETURN                            <200>
3097 REM ++++++++++++++++++++++++             <183>
3098 REM + AUFPRALL VON SCHUSS +              <141>
3099 REM ++++++++++++++++++++++++             <185>
3100 POKE SI+10,3:GOSUB 60700:IF PEEK(4915
     2+E+S+40*Y1)<>51 THEN 3150               <123>
3102 POKE 49152+E+S+40*Y1,38:FOR T=0 TO 10
     0:NEXT:POKE 49152+E+S+40*Y1,32:E=0       <168>
3103 SC=SC+6:GOSUB 61200:RETURN               <189>
3110 POKE SI+10,3:GOSUB 60700:IF PEEK(4915
     2+X1+40*E+40)<>51 THEN 3170              <095>
3111 POKE 49152+X1+40*E+40,38                 <083>
3112 FOR T=0 TO 110:NEXT:POKE 49152+X1+40*
     E+40,32:E=0:SC=SC+8:GOSUB 61200:RETUR
     N                                        <007>
3150 PO=PEEK(49152+E+S+40*Y1):POKE 49152+E
     +S+40*Y1,38                              <170>
3160 FOR T=0 TO 100:NEXT:POKE 49152+E+S+40
     *Y1,PO:E=0                               <208>
3165 RETURN                                   <175>
3170 PO=PEEK(49152+X1+40*E+40):POKE 49152+
     X1+40*E+40,38                            <177>
3180 FOR T=0 TO 100:NEXT:POKE 49152+X1+40*
     E+40,PO:E=0:RETURN                       <216>
3197 REM ↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑         <155>
3198 REM ↑ KOLLISION HUBSCHRAUBER ↑           <023>
3199 REM ↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑         <157>
3200 GOSUB 60200:POKE SI+18,0:POKE V+2,0:X
     2=0:Y2=INT(RND(1)*50)+90:HB=HB-1         <028>
3201 POKE V+3,Y2:O=PEEK(V+31):JS=PEEK(V+30
     )                                        <103>
3202 FOR T=0 TO 100:P=INT(RND(1)*15):POKE
     V+39,P:NEXT:POKE V+39,3                  <037>
3203 IF HB<=0 THEN POKE V+21,0:PRINT"{HOME
     ,8DOWN,12RIGHT}G A M E{2SPACE}O V E R
     ":GOTO 8000                              <027>
3210 X=250:Y=94:POKE V,X:POKE V+1,Y:O=PEEK
     (V+31):GOSUB 60300:GOTO 1009             <133>
3300 X1=INT(X/8-4):Y1=INT(Y/8-5)              <069>
3310 FOR E=X1 TO 0 STEP-1:GOSUB 61400         <145>
3315 Y1=Y1+.5                                 <136>
3316 IF Y1>=24 THEN Y1=24:RETURN              <172>
3319 PO=PEEK(49152+E+40*INT(Y1))              <192>
3320 POKE 49152+E+40*INT(Y1),67               <207>
3321 POKE 49152+E+40*INT(Y1),PO               <117>
3330 IF PEEK(49152+E-1+40*INT(Y1))<>32 THE
     N S=-1:GOTO 3100                         <249>
3340 NEXT E:RETURN                            <206>
3400 X1=INT(X/8+4):Y1=INT(Y/8-5)              <167>
3410 FOR E=X1 TO 39 STEP 1:GOSUB 61400        <017>
3415 Y1=Y1+.5                                 <238>
3416 IF Y1>=24 THEN Y1=24:RETURN              <018>
3419 PO=PEEK(49152+E+40*INT(Y1))              <038>
3420 POKE 49152+E+40*INT(Y1),69               <118>
3421 POKE 49152+E+40*INT(Y1),PO               <219>
3430 IF PEEK(49152+E+1+40*INT(Y1))<>32 THE
     N S=1:GOTO 3100                          <050>
3440 NEXT E:RETURN                            <050>
3997 REM ←←←←←←←←←←←←←←←←←←←←←←←←             <231>
```

```
3998 REM ← BILD TANKEN AUFBAUEN ←                <099>
3999 REM ←←←←←←←←←←←←←←←←←←←←←←←←←               <233>
4000 GOSUB 60500:X2=0:Y2=90:POKE V+2,X2:PO
     KE V+3,Y2                                   <206>
4001 X=250:Y=94:BI=1                             <177>
4002 TR=49152:C=40:GOSUB 4530                    <020>
4009 REM ↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑           <047>
4010 REM ↑ BILDSCHIRMABFRAGE TANKEN ↑            <123>
4011 REM ↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑           <049>
4500 IF BI=0 THEN 4000                           <148>
4501 IF B9=0 THEN:IF EN>30 AND HU<5 THEN B
     9=1:POKE V+21,3:GOSUB 61570:X3=20           <116>
4502 ON B9 GOSUB 61300                           <204>
4503 IF Y<=175 AND KL=0 THEN RETURN              <173>
4504 IF X>140 AND X<175 AND KL=0 THEN KL=1
     :GOSUB 4511:IF LD=1 THEN GOTO 62000         <192>
4505 PU=.05:IF Y>214 AND KL=1 THEN H=160:O
     T=1:AB=AC:PU=-1:GOSUB 60800                 <107>
4506 IF Y>218 AND KL=1 THEN Y=Y-AB-1             <229>
4507 IF Y<=170 AND KL=1 THEN KL=0:GOSUB 45
     30:RETURN                                   <142>
4508 IF Y>214 AND KL=1 AND EN>=35 THEN EN=
     35:H=32:POKE SI+18,0                        <150>
4509 IF Y<=214 AND KL=1 THEN POKE SI+18,0        <191>
4510 RETURN                                      <250>
4511 FOR T=18 TO 23                              <178>
4512 FOR I=15 TO 24                              <156>
4515 POKE TR+I+40*T,32                           <130>
4520 NEXT:NEXT                                   <205>
4521 PRINT"{HOME,22DOWN}"                        <229>
4522 PRINT"{RED,10SPACE,WHITE}>{RED}####{B
     LUE,SPACE}00000000{SPACE,RED}####{WHI
     TE}?                                        <141>
4523 PRINT"{RED}$$$$$$$$$$$######{GREEN,2SPA
     CE}TJ{2SPACE}TJ{2SPACE,RED}######$$$$
     $$$${HOME,LIG.BLUE}                         <246>
4529 RETURN                                      <013>
4530 PRINT"{HOME,17DOWN}"                        <034>
4531 PRINT"{15SPACE,WHITE}%%%%%%%%%%             <134>
4532 PRINT"{14SPACE,WHITE}>###{RED}/012{WH
     ITE}###?                                    <229>
4533 PRINT"{13SPACE,WHITE}>{RED}#{GREEN}''
     ''''''''{RED}#{WHITE}?                      <130>
4534 PRINT"{12SPACE,WHITE}>{RED}##{GREEN}'
     '''''''''{RED}##{WHITE}?                    <220>
4536 PRINT"{11SPACE,WHITE}>{RED}###{GREEN}
     ''''''''''''{RED}###{WHITE}?                <141>
4537 PRINT"{10SPACE,WHITE}>{RED}####{GREEN
     }''''''''''''{RED}####{WHITE}?              <224>
4538 PRINT"{RED}$$$$$$$$$$######{GREEN}''''
     ''''''{RED}######$$$$$$$$${HOME,LIG.BL
     UE}                                         <169>
4560 KL=0:RETURN                                 <135>
4997 REM ←←←←←←←←←←←←←←←←←←←←←←←←←               <215>
4998 REM ← BILD TUNNEL AUFBAUEN ←                <083>
4999 REM ←←←←←←←←←←←←←←←←←←←←←←←←←               <217>
5000 IF BI=0 THEN H=32:GOTO 5010                 <139>
5005 GOTO 5300                                   <199>
5010 GOSUB 60500:TI$="000000"                    <026>
5011 X=250:Y=94:BI=1:PX=45:PY=46                 <013>
5019 PRINT"{HOME,3DOWN,RED}"                     <183>
5020 PRINT"{4SPACE}>66776567565677567656667
     878776767657                                <055>
5030 FOR T=0 TO 3:PRINT"{4SPACE}()":NEXT         <066>
5035 PRINT"{4SPACE}(){8SPACE}6786876887678
     768676787867                                <137>
5040 FOR T=0 TO 2:PRINT"{RED,4SPACE}(){27S
     PACE,BLUE}++":NEXT                          <052>
5045 PRINT"{WHITE,SPACE}(67867887876767676
     668787{8SPACE,BLUE}++                       <064>
5050 PRINT"{WHITE,SPACE}(9:{YELLOW}))))))
     ))))))))){WHITE}9;9:{8SPACE,BLUE}++"        <222>
5085 PRINT"{WHITE,SPACE}({10SPACE,PURPLE}3
     33{8SPACE}3{9SPACE,BLUE}++                  <003>
5090 PRINT"{WHITE,SPACE}({11SPACE,PURPLE}3
     {9SPACE}3{9SPACE,BLUE}++                    <010>
5095 PRINT"{WHITE,SPACE}({21SPACE,PURPLE}3
     {9SPACE,BLUE}++                             <186>
5096 PRINT"{WHITE,SPACE}({9SPACE,PURPLE}33
     333{7SPACE,PURPLE}3{9SPACE,BLUE}++          <078>
5100 PRINT"{WHITE,SPACE}788{CYAN}3333333{W
     HITE}8887878787878787{8SPACE,BLUE}++        <101>
5110 PRINT"{WHITE,SPACE};:9{CYAN}3333333{W
     HITE};:9:9:;9:9:;9:{8SPACE,BLUE}++          <215>
5120 PRINT"{12SPACE,LIG.BLUE,3SPACE}NM{16S
     PACE,BLUE}++                                <164>
5125 PRINT"{12SPACE,LIG.BLUE,2SPACE}N{2SPA
     CE}M{15SPACE,BLUE}++                        <169>
5130 PRINT"{12SPACE,LIG.BLUE,SPACE}N{4SPAC
     E}M{14SPACE,BLUE}++                         <174>
5135 PRINT"{RED}$$$$$$$$$$$${LIG.BLUE}#{RE
     D}$$$$$${LIG.BLUE}#{RED}$$$$$$$$$$$$$$
     {BLUE}++{LIG.BLUE,HOME}                     <009>
5297 REM ++++++++++++++++++++++++++++            <069>
5298 REM + BILDSCHIRMABFRAGE TUNNEL +            <091>
5299 REM ++++++++++++++++++++++++++++            <071>
5300 POKE TR+Q+C*17,PX                           <250>
5320 POKE TR+Q+C*17,PY:IF Y<170 THEN RETUR
     N                                           <168>
5340 GE=INT(RND(1)*15):IF GE>10 THEN GOSUB
      60000                                      <035>
5345 IF X<140 AND X>120 AND Y>174 AND Y<17
     8 THEN PX=32:PY=32:GOSUB 61100:GOSUB
     61200                                       <201>
5350 IF TI$>"000005"THEN GOSUB 60070             <251>
5407 IF Y<=214 THEN POKE SI+18,0                 <078>
5410 RETURN                                      <134>
5497 REM ↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑         <152>
5498 REM ↑ BILDSCHIRM PANZER AUFBAUEN ↑          <140>
5499 REM ↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑         <154>
5500 IF BI=0 THEN H=32:GOTO 5510                 <005>
5505 GOTO 5700                                   <255>
5510 GOSUB 60500                                 <100>
5520 X=250:Y=94:BI=1:B5=-2:POKE V+5,223:PO
     KE V+7,223:X5=50:X6=120:POKE V+21,13        <134>
5521 POKE V+4,X5:POKE V+6,X6:GOSUB 5800          <176>
5530 PRINT"{HOME,3DOWN,BLUE}":B6=2:Y7=20:G
     9=1                                         <006>
5540 PRINT"76787878787878686888786878687 86
     87868786"                                   <056>
5545 PRINT":;:;:;9:;9;9:9:;9:;9{19SPACE}"       <251>
5547 PRINT"+){16SPACE}(+"                        <228>
5550 FOR T=0 TO 2:PRINT"+){16SPACE}(+{CYAN
     }333333{BLUE}":NEXT                         <031>
5555 PRINT"+){8SPACE}+{7SPACE}(+{CYAN}3333
     33333{BLUE}7786876787"                      <221>
5560 FOR T=0 TO 3:PRINT"{CYAN,SPACE}3{BLUE
     ,8SPACE}+{7SPACE}(+{CYAN}333333333{BL
     UE}+)":NEXT                                 <138>
5565 FOR T=0 TO 5:PRINT"{BLUE}+){8SPACE,BL
     UE}+{7SPACE,CYAN}333{2SPACE}333333{BL
     UE}+)":NEXT                                 <030>
5570 FOR T=0 TO 2:PRINT"{RED,18SPACE}(+{CY
     AN}333333333{RED}+)":NEXT                   <089>
5575 PRINT"{GREEN}$$$$$$$$$$$$$$$$$$$$$$$$$
     $$$$$$$$$$$$$$${LIG.BLUE,HOME}"             <144>
5697 REM ++++++++++++++++++++++++++++            <217>
5698 REM + BILDSCHIRMABFRAGE PANZER +            <142>
5699 REM ++++++++++++++++++++++++++++            <219>
5700 IF X>168 THEN RETURN                        <014>
5710 IF X5>138 THEN B5=3:POKE 50170,24           <059>
5715 IF X5<30 THEN B5=-2:POKE 50170,23           <200>
5720 IF X6>138 THEN B6=2:POKE 50171,24           <077>
5730 IF X6<30 THEN B6=-3:POKE 50171,23           <027>
5740 X5=X5-B5:X6=X6-B6                           <065>
5750 POKE V+4,X5:POKE V+6,X6                     <175>
5755 JS=PEEK(V+30):IF JS=5 OR JS=9 OR JS=1
     3 THEN 3200                                 <202>
5765 ON G9 GOSUB 5820,5900                       <110>
5770 RETURN                                      <240>
5800 X7=INT(X5/8-1):Y7=21                        <049>
5810 RETURN                                      <024>
5811 X8=INT(X6/8-1):Y8=21                        <070>
5815 RETURN                                      <029>
5820 Y7=Y7-1                                     <047>
5835 PO=PEEK(49152+X7+40*Y7-40)                  <118>
5840 POKE 49152+X7+40*Y7,90                      <185>
5850 IF PO<>32 THEN POKE 49152+X7+40*Y7-40
     ,PO:POKE 49152+X7+40*Y7,32:G9=2:GOSUB
      5811                                       <088>
5860 POKE 49152+X7+40*Y7,32                      <197>
5870 RETURN                                      <084>
5900 Y8=Y8-1                                     <202>
5920 PO=PEEK(49152+X8+40*Y8-40)                  <250>
5930 POKE 49152+X8+40*Y8,90                      <026>
5940 IF PO<>32 THEN POKE 49152+X8+40*Y8-40
     ,PO:POKE 49152+X8+40*Y8,32:G9=1:GOSUB
      5800                                       <163>
5950 POKE 49152+X8+40*Y8,32                      <038>
5960 RETURN                                      <176>
5997 REM ↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑         <144>
```

Listing »APOCALYPSE NOW« (Fortsetzung)


Ungla


64ER ONLINE

```
5998 REM ↑ BILDSCHIRM SPINNE AUFBAUEN ↑      <107>
5999 REM ↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑      <146>
6000 IF BI=0 THEN H=32:GOTO 6020              <188>
6010 GOTO 6300                                <196>
6020 GOSUB 60500:TI$="000000":V=53248:POKE
      50172,25:POKE V+8,50:POKE V+9,157      <204>
6030 X=250:Y=110:BI=1:POKE V+23,16:POKE V+
     29,19:POKE V+21,17                       <222>
6040 POKE V,X:POKE V+1,Y:O=PEEK(V+31)         <055>
6050 PRINT"{HOME,3DOWN,BLUE}"                 <201>
6060 PRINT"{12SPACE,WHITE}6787878678687868
     76878677867                              <041>
6070 PRINT"{12SPACE,WHITE}+;9:;9:9;9;9:9;+
     9:;:;B;9:;9                              <006>
6080 FOR T=0 TO 1:PRINT"{12SPACE,BLUE}+){1
     2SPACE}(+":NEXT                          <139>
6090 FOR T=0 TO 2:PRINT"{12SPACE}+){15SPAC
     E}":NEXT                                 <006>
6100 PRINT"{SPACE,WHITE}767687878 68+         <104>
6110 PRINT"{BLUE}(+++++++++++{15SPACE}+) ↑
     ACHTUNG↑                                 <163>
6120 PRINT"(++{7SPACE}++){13SPACE}(+)         <219>
6130 PRINT"(++{7SPACE}++{15SPACE}+)           <002>
6140 PRINT"(+{8SPACE,CYAN}333{BLUE,14SPACE
     }+)                                      <008>
6150 PRINT"(+{8SPACE,CYAN}333{BLUE,14SPACE
     }+)                                      <020>
6160 PRINT"(++{7SPACE}++{15SPACE}+)           <034>
6170 PRINT"(++{7SPACE}++){13SPACE}(+)         <015>
6180 PRINT"(++++++++++7{14SPACE,BLUE}+)       <119>
6190 PRINT"{WHITE,SPACE}::;9:;:9;9:+{14SPA
     CE,BLUE}+)                               <196>
6200 PRINT"{12SPACE}+){4SPACE,PURPLE}+({7S
     PACE}+)                                  <148>
6210 PRINT"{12SPACE,BLUE}+){4SPACE,PURPLE}
     +({GREEN}PPPPPPP{PURPLE}+)               <099>
6220 PRINT"{12SPACE,BLUE}+){4SPACE,PURPLE}
     +({2SPACE,CYAN}/012{SPACE,PURPLE}+)      <211>
6230 PRINT"{GREEN}$$$$$$$$$$$${BLUE}+{GREE
     N}$$$$${PURPLE}+{GREEN}$$$$$$$${PURPL
     E}+{GREEN}$$$$$$$$$${LIG.BLUE,HOME}"     <204>
6297 REM +++++++++++++++++++++++++++++++      <053>
6298 REM + BILDSCHIRMABFRAGE SPINNE +         <093>
6299 REM +++++++++++++++++++++++++++++++      <055>
6300 IF TI$>"000007"THEN GOSUB 6400           <147>
6305 IF TI$>"000010"THEN GOSUB 6420           <216>
6310 PU=.05:IF Y>206 AND X>180 AND X<190 T
     HEN GOSUB 6440                           <227>
6320 IF Y<=206 THEN POKE SI+18,0              <103>
6325 O=PEEK(V+31):IF O=16 THEN GOSUB 6550     <050>
6399 RETURN                                   <105>
6400 PRINT"{HOME,13DOWN,13RIGHT,GREEN}FFFF
     FFFFFFFFF{LIG.BLUE}"                     <002>
6405 O=PEEK(V+31):IF O=1 THEN 3200            <108>
6410 PRINT"{HOME,13DOWN,13RIGHT,13SPACE}"     <039>
6415 RETURN                                   <123>
6420 PRINT"{HOME,18DOWN,13RIGHT,GREEN}FFFF
     FFFFFFFFF{LIG.BLUE}"                     <036>
6425 O=PEEK(V+31):IF O=1 THEN 3200            <128>
6430 PRINT"{HOME,18DOWN,13RIGHT,13SPACE}"     <185>
6435 TI$="000000":RETURN                      <021>
6440 IF HA=1 THEN RETURN                      <225>
6441 PU=.05:OT=1:H=160:PU=-1:GOSUB 60800:I
     F EN>35 THEN EN=35:HA=1:H=32:PU=.05:O
     T=1                                      <136>
6450 IF Y>209 THEN Y=Y-AB-1                   <202>
6460 RETURN                                   <168>
6550 FOR T=0 TO 180:POKE 53280,T:POKE 5328
     1,T:GOSUB 60700:NEXT:POKE 53280,0        <104>
6552 POKE 53281,0:LD=1:SC=SC+1000:GOSUB 61
     200                                      <052>
6560 POKE V+21,1:RETURN                       <019>
6697 REM +++++++++++++++++++                  <158>
6698 REM + PUNKTE WERTUNG +                   <009>
6699 REM +++++++++++++++++++                  <160>
8000 FOR T=0 TO 1500:NEXT:PRINT"{CLR,4DOWN
     ,6RIGHT}NOCH EINMAL [J] "                <048>
8002 PRINT"{HOME,10DOWN,10RIGHT}SCORE{RVSO
     N,SPACE}";SC                             <176>
8003 IF SC>AS THEN AS=SC                      <084>
8004 PRINT"{HOME,12DOWN,8RIGHT}HIGHSCORE{S
     PACE,RVSON}";AS                          <091>
8010 POKE 198,0:WAIT 198,1:GET A$:IF A$="J
     "THEN 9000                               <135>
8015 POKE 648,4:SYS 65126                     <233>
8020 PRINT"{CLR,SPACE}DANKE!!! GOOD BYE !!
     !!!!!!!":END                             <144>
9000 BI=0:B9=0:EN=0:HU=0:PU=0:AB=0:KL=0:H=
     0:PX=0:PY=0:GE=0:PO=0:G9=0               <046>
9010 B=0:HA=0:LD=0:SC=0:AC=0:POKE V+23,0:P
     OKE V+29,0:GOTO 998                      <004>
10000 REM ↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑          <127>
10001 REM ↑ ANFANGSSPRITE BEWEGEN ↑           <213>
10002 REM ↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑          <129>
50000 POKE V,X:POKE V+1,Y                     <114>
50010 X1=INT(X/8-4):Y1=INT(Y/8-5)             <042>
50020 FOR E=X1 TO 0 STEP-1                    <169>
50030 POKE 49152+E+40*Y1,66:POKE 49152+E+4
      0*Y1,32                                 <100>
50040 IF PEEK(49152+E-1+40*Y1)<>32 THEN PO
      KE 49152+E-1+40*Y1,38:GOTO 50070        <025>
50050 NEXT E:RETURN                           <177>
50070 FOR T=0 TO 100:NEXT:POKE 49152+E-1+4
      0*Y1,32:X=X-3:E=0:GOTO 50000            <233>
60000 POKE TR+Q+C*17,32:Q=Q-INT(RND(1)*3):
      Q=Q+INT(RND(1)*3)                       <099>
60010 IF Q<11 THEN Q=11                       <122>
60020 IF Q>14 THEN Q=14                       <061>
60060 RETURN                                  <171>
60070 OT=OT*-1:TI$="000000"                   <167>
60075 PRINT"{HOME,20DOWN}"                    <103>
60080 IF OT=-1 THEN FOR T=0 TO 2:PRINT"{13
      RIGHT,6SPACE}":NEXT:RETURN              <182>
60085 PRINT"{LIG.BLUE,13RIGHT,2SPACE}NM{2S
      PACE}"                                  <064>
60086 PRINT"{13RIGHT,SPACE}N{2SHIFT-SPACE}
      M "                                     <168>
60087 PRINT"{13RIGHT}N{4SPACE}M{LIG.BLUE}"    <047>
60090 RETURN                                  <201>
60099 REM GERAEUSCHE                          <062>
60100 SI=54272:POKE SI+7,200:POKE SI+8,0:P
      OKE SI+9,0:POKE SI+10,3:POKE SI+11,0    <129>
60110 POKE SI+12,0:POKE SI+13,15*16+15:POK
      E SI+21,0:POKE SI+22,150:POKE SI+23,
      243                                     <185>
60120 POKE SI+24,31:POKE SI+11,65             <044>
60130 RETURN                                  <241>
60200 POKE SI+11,0                            <080>
60201 POKE SI+1,5:POKE SI+4,0:POKE SI+5,11
      :POKE SI+6,0                            <190>
60202 POKE SI+4,131                           <096>
60210 RETURN                                  <067>
60300 IF EN<=1 THEN EN=15:IF B>1 THEN EN=3
      5                                       <161>
60301 PRINT"{HOME,3DOWN,10SPACE,BLUE,26SPA
      CE,LIG.BLUE}"                           <214>
60302 PRINT"{HOME,DOWN,25RIGHT}LEBEN{SPACE
      ,RVSON}"HB                              <120>
60303 PRINT"{HOME,2DOWN,2RIGHT,YELLOW}ENER
      GY{LIG.BLUE}":FOR T=0 TO EN:POKE 491
      52+T+40*3,160:NEXT                      <178>
60304 PRINT"{HOME,9RIGHT,15SPACE}"            <054>
60310 X=250:Y=94:AB=AC:H=32:GOSUB 61200       <206>
60315 IF B=4 THEN Y=110                       <081>
60320 RETURN                                  <177>
60500 PRINT"{CLR,3DOWN,10SPACE,BLUE,26SPAC
      E,LIG.BLUE}"                            <167>
60501 PRINT"{HOME,2DOWN,2RIGHT,YELLOW}ENER
      GY{LIG.BLUE}"                           <164>
60502 FOR T=0 TO EN:POKE 49152+T+40*3,160:
      NEXT:GOSUB 61200                        <187>
60550 RETURN                                  <153>
60700 SI=54272:POKE SI+4,0:POKE SI+1,10:PO
      KE SI+5,12:POKE SI+6,0:POKE SI+4,131    <070>
60710 RETURN                                  <059>
60720 SI=54272:POKE SI+4,0:POKE SI+1,20:PO
      KE SI+5,12:POKE SI+6,0:POKE SI+4,131    <092>
60730 RETURN                                  <079>
60800 SI=54272:POKE SI+18,0:POKE SI+19,0:P
      OKE SI+20,15*16                         <053>
60810 POKE SI+14,200:POKE SI+15,39:POKE SI
      +18,17                                  <133>
60820 RETURN                                  <169>
60900 SI=54272:POKE SI+18,0:POKE SI+19,0:P
      OKE SI+20,15*16+2                       <182>
60910 POKE SI+14,200:POKE SI+15,59:POKE SI
      +18,17                                  <009>
60920 RETURN                                  <013>
61000 OT=OT*-1:IF OT=-1 THEN PRINT"{HOME,9
      RIGHT,15SPACE}":RETURN                  <205>
61001 PRINT"{HOME,CYAN,13RIGHT}LOW ON{SPAC
      E,RED}FUEL{LIG.BLUE}"                   <093>
61002 RETURN                                  <097>
61100 SC=SC+100:PRINT"{HOME,23RIGHT,15DOWN
```

```
        ,RED}+{DOWN,LEFT}+{DOWN,LEFT}+{DOWN,
        LEFT}+{LIG.BLUE}"                    <154>
61120 RETURN                                 <215>
61200 PRINT"{HOME,25RIGHT,2DOWN,RVSON,GREE
        N}SCORE ";SC;"{RVOFF,LIG.BLUE}"      <065>
61210 RETURN                                 <051>
61297 REM ++++++++++++++++++++++++++++       <081>
61298 REM + FEINDHUBSCHRAUBER BEWEGEN +      <019>
61299 REM ++++++++++++++++++++++++++++       <083>
61300 IF X>X2 THEN B2=+5:M4=22               <114>
61310 IF X<X2 THEN B2=-5:M4=21               <160>
61320 IF Y>Y2 THEN C2=+3                     <253>
61330 IF Y<Y2 THEN C2=-3                     <047>
61335 X2=X2+B2:Y2=Y2+C2                      <183>
61340 POKE V+2,X2:POKE V+3,Y2                <097>
61341 O=PEEK(V+31)                           <162>
61342 JS=PEEK(V+30)                          <017>
61350 IF JS=3 THEN 3200                      <007>
61351 IF O=2 THEN 61500                      <127>
61352 JS=PEEK(V+30)                          <027>
61353 IF HU>3 THEN GOSUB 61530               <193>
61354 O=PEEK(V+31)                           <175>
61355 POKE 50169,M4                          <121>
61360 RETURN                                 <201>
61400 IF B=3 THEN 61430                      <114>
61401 IF B=4 THEN 61470                      <163>
61402 IF HU>4 THEN RETURN                    <034>
61410 O=PEEK(V+31):IF O=2 THEN 61500         <167>
61420 RETURN                                 <005>
61430 O=PEEK(V+31):IF O=12 THEN GOSUB 6145
        0:POKE V+21,1:SC=SC+200:GOSUB 61200:
        G9=0                                 <200>
61440 RETURN                                 <027>
61450 GOSUB 60720:FOR T=0 TO 100:POKE V+38
        ,T:NEXT:RETURN                       <240>
61470 O=PEEK(V+31):IF O=16 THEN:GOSUB 6550   <034>
61480 RETURN                                 <067>
61500 HU=HU+1:SC=SC+100:GOSUB 61200          <065>
61501 GOSUB 60700                            <018>
61502 FOR T=0 TO 100:POKE V+40,T:NEXT:X2=0
        :POKE V+2,X2:Y2=INT(RND(1)*50)+90    <146>
61503 POKE V+3,Y2                            <041>
61504 IF HU>4 THEN POKE V+21,1:B9=0          <122>
61510 E=0:RETURN                             <190>
61530 X3=X3+1                                <189>
61535 IF X3>X/8 THEN GOSUB 61570             <145>
61536 IF X3>39 THEN GOSUB 61570              <106>
61540 POKE 49152+X3+40*INT(Y3),68            <177>
61545 O=PEEK(V+31):IF O=1 THEN POKE 49152+
        X3+40*INT(Y3),32:GOTO 3200           <184>
61550 POKE 49152+X3+40*INT(Y3),32            <217>
61560 RETURN                                 <147>
61570 X3=INT(X2/8+4):Y3=INT(Y2/8-5)          <162>
61580 RETURN                                 <167>
61997 REM ←←←←←←←←←                          <068>
61998 REM ← MUSIK ←                          <074>
61999 REM ←←←←←←←←←                          <070>
62000 FOR I=54272 TO 54296:POKE I,0:NEXT:P
        OKE 54296,15:POKE 54275,3            <170>
62110 H1=54272:H2=H1+1:AD=54277:SR=AD+1:DR
        =AD-1:W1=64:W2=32                    <112>
62120 POKE AD,72:POKE SR,122:POKE AD+7,43:
        POKE SR+7,187:GOTO 62250             <160>
62130 DIM F$(18),FL(18),FH(18):FOR I=0 TO
        18                                   <079>
62140 READ F$(I),F:FH(I)=INT(F/256):FL(I)=
        F-FH(I)*256:NEXT                     <236>
62150 DIM N1$(200),D1(200),N2$(200),D2(200
        ):I=0                                <159>
62160 READ N1$(I),D1(I):IF D1(I)>0 THEN I=
        I+1:GOTO 62160                       <141>
62170 I=0                                    <124>
62180 READ N2$(I),D2(I):IF D2(I)>0 THEN I=
        I+1:GOTO 62180                       <205>
62200 DIM L1(200),H1(200),L2(200),H2(200):
        I=0                                  <202>
62210 FOR J=0 TO 18                          <107>
62220 IF F$(J)=N1$(I)THEN L1(I)=FL(J):H1(I
        )=FH(J)                              <121>
62230 IF F$(J)=N2$(I)THEN L2(I)=FL(J):H2(I
        )=FH(J)                              <169>
62240 NEXT:I=I+1:IF N1$(I)<>""THEN 62210     <037>
62245 RETURN                                 <070>
62250 SP=VAL("2")                            <248>
62260 IF SP<1 OR SP>4 THEN 62250             <214>
62270 S1=0:S2=0:C1=0:C2=0:POKE H1,L1(S1):P
        OKE H2,H1(S1)                        <199>
62280 POKE H1+7,L2(S2):POKE H2+7,H2(S2):PO
        KE DR,W1 OR 1:POKE DR+7,W2 OR 1      <208>
62290 C1=C1+1:IF C1>(D1(S1)*SP/2)THEN POKE
        DR,W1                                <053>
62300 IF C1<(D1(S1)*SP)THEN 62320            <032>
62310 C1=0:S1=S1+1:POKE DR,W1 OR 1:POKE H1
        ,L1(S1):POKE H2,H1(S1)               <182>
62320 C2=C2+1:IF C2>(D2(S2)*SP/2)THEN POKE
        DR+7,W2                              <183>
62330 IF C2<(D2(S2)*SP)THEN 62350            <183>
62340 C2=0:S2=S2+1:POKE DR+7,W2 OR 1:POKE
        H1+7,L2(S2):POKE H2+7,H2(S2)         <218>
62350 IF D1(S1)>0 THEN 62290                 <211>
62360 POKE V+21,0:GOTO 8000                  <129>
62370 DATA D3,2408,E3,2703,F#3,3034,G3,321
        5                                    <209>
62380 DATA A3,3608,H3,4050,C#4,4547,D4,481
        7                                    <052>
62390 DATA E4,5407,F#4,6069,G4,6403,A4,721
        7                                    <212>
62400 DATA H4,8101,C#5,9094,D5,9634,E5,108
        14                                   <019>
62410 DATA F#5,12139,G5,12860,A5,14435       <018>
62420 REM  1.STIMME                          <131>
62430 DATA D5,3,D5,1,D5,1,E5,1,D5,1,C#5,1,
        H4,4,H4,4,C#5,1,C#5,1                <153>
62440 DATA D5,1,E5,3,D5,1,E5,1,F#5,2,F#5,2
        ,E5,1,A4,3                           <139>
62450 DATA D5,3,D5,1,D5,1,E5,1,D5,1,C#5,1,
        H4,4,H4,4,C#5,1,C#5,1                <173>
62460 DATA D5,1,E5,3,D5,1,E5,1,F#5,2,F#5,2
        ,E5,1,A4,3                           <159>
62470 DATA F#5,2,A5,2,G5,2,F#5,1,D5,1,E5,1
        ,F#5,2,E5,5                          <036>
62480 DATA F#5,3,E5,1,D5,2,D5,1,H4,1,E5,1,
        F#5,2,E5,3,D5,1,E5,1                 <062>
62490 DATA F#5,2,A5,2,G5,2,F#5,1,D5,1,E5,1
        ,C#5,1,H4,1,A4,5                     <230>
62500 DATA D5,3,C#5,1,H4,1,C#5,1,D5,2,E5,1
        0                                    <022>
62510 DATA F#5,2,E5,6,F#5,2,E5,2,D5,1,E5,1
        ,F#5,2,A5,2,G5,2,F#5,1,D5,3,F#5,2,E5
        ,6                                   <112>
62520 DATA F#5,2,E5,6,F#5,2,E5,2,D5,1,E5,1
        ,F#5,2,A5,2,G5,2,F#5,1,D5,1          <221>
62530 DATA E5,1,C#5,1,H4,1,A4,3,H4,1,C#5,1   <020>
62540 DATA D5,3,D5,1,D5,1,E5,1,D5,1,C#5,1,
        H4,4,H4,4,C#5,1,C#5,1                <009>
62550 DATA D5,1,E5,3,D5,1,E5,1,F#5,2,F#5,2
        ,E5,1,A4,3                           <251>
62560 DATA D5,3,D5,1,D5,1,E5,1,D5,1,C#5,1,
        H4,4,H4,4,C#5,1,C#5,1                <029>
62570 DATA D5,1,E5,3,D5,1,E5,1,F#5,2,F#5,2
        ,E5,1,D5,7                           <052>
62580 DATA,-1                                <196>
62590 REM  2.STIMME                          <055>
62600 DATA D3,4,D3,4,G3,4,H3,2,D3,2,A3,4,A
        3,4,D3,4,A3,2,E4,1,A3,1              <169>
62610 DATA D3,4,D3,4,G3,4,H3,2,D3,2,A3,4,A
        3,4,D3,4,A3,2,E4,1,A3,1              <179>
62620 DATA D3,4,D3,4,A3,4,A3,4,H3,4,G3,4,A
        3,4,A3,4                             <211>
62630 DATA D3,4,D3,4,A3,4,A3,4,H3,4,G3,4,A
        3,4,A3,3                             <219>
62640 DATA A3,1,A3,7,A3,1,A3,8,D3,4,D3,4,A
        3,4,A3,3                             <109>
62650 DATA A3,1,A3,7,A3,1,A3,8,D3,4,D3,4,A
        3,4,A3,4                             <121>
62660 DATA D3,4,D3,4,G3,4,H3,2,D3,2,A3,4,A
        3,4,D3,4,A3,2,E4,1,A3,1              <229>
62670 DATA D3,4,D3,4,G3,4,H3,2,D3,2,A3,4,A
        3,4,D3,4,D3,8                        <144>
62680 DATA,-1                                <040>
62997 REM +++++++++++++++                    <140>
62998 REM + GRAVITATION +                    <016>
62999 REM +++++++++++++++                    <142>
63000 PRINT CHR$(142):POKE V+21,0            <166>
63001 PRINT"{CLR,10DOWN,8RIGHT,RVSON}GRAVI
        TATION{2SPACE}[ 1 £ 3 ]"             <218>
63010 POKE 198,0:WAIT 198,1:GET A$:IF A$="
        1"THEN AC=.1                         <143>
63020 IF A$="2"THEN AC=.5                    <087>
63030 IF A$="3"THEN AC=1                     <245>
63040 IF A$<"1"OR A$>"3"THEN 63010           <155>
63050 RETURN                                 <113>
```

© 64'er   **Listing »APOCALYPSE NOW« (Schluß)**

Spring Vogel, spring

Kennen Sie Jumpman, Miner 2049 oder Mister Robot and his Factory? Dann haben Sie sich sicherlich geärgert, daß Sie nie die letzten Bilder erreicht haben. Oder Sie haben sich über die eintönigen Spielszenen geärgert. Mit dem Spring-Vogel passiert so etwas nicht! Hier können Sie sich Ihre Bilder auch selber machen.

Tatsächlich kann man Spring-Vogel fast als Spielgenerator bezeichnen. Mit Spring-Vogel gelingt es Ihnen, sich jedes beliebige Spielfeld aufzubauen. Das klingt sehr vielversprechend, und das ist es auch. Spring-Vogel ist ein Vertreter der Jump and Run-Kategorie. Damit lassen sich also keine Schieß- oder Abenteuerspiele erzeugen, aber innerhalb der Spring- und Laufgruppe bleibt kein Wunsch offen. Worum handelt es sich bei Spring-Vogel nun eigentlich? Zunächst die Story.

Ein heftiger Sturm hat einen Vogel — unseren Helden — mitsamt seinen Eiern aus dem Nest geweht. Durch den harten Aufprall auf die Erde hat er sich seine Flügel gebrochen. Er kann also nicht mehr richtig fliegen, sondern nur noch auf dem Boden laufen, hüpfen und springen. Die Aufgaben des Vogels (die genaue Klassifizierung bleibt Ihnen überlassen) besteht nun darin, in einem Labyrinth aus Aufzügen, Transportbändern, Seilen, Einbahnstraßen, Trampolinen, Rutschbahnen, Gummiwänden, magischen Flügeln, gemeinen Vogelfallen und mißgestimmten Monstern alle Eier wieder einzusammeln. Gelingt es dem Vogel, die Eier in einer Spielszene aufzunehmen, muß er mit der nächsten Überraschung fertig werden; es waren nicht alle. Das nächste vom Winde verwehte Bild erwartet ihn mit weiteren schwierigen Aufgaben.

Die Anzahl der Torturen für unseren leidgeprüften Helden bestimmen Sie selbst. Doch Vorsicht, unser Vogel verfügt nicht wie eine Katze über sieben, sondern »nur« über sechs Leben. Und diese sechs Leben sind schnell, durch Kontakt mit den Monstern oder Verzehren einer roten Tollkirsche, ausgehaucht.

Der Sturz über mehr als vier Etagen ist ebenfalls für den flugunfähigen Vogel lebensbedrohend. Es sei denn, er findet die von seinem Tierschützer verstreuten »magischen Flügel«. Ausgestattet mit deren zauberhaften Fähigkeiten kann er in einem ökologischen Flecken beliebig umherfliegen. Durch Betätigen des Feuerknopfes wird er wieder in seinen behinderten Zustand zurückversetzt, erlangte allerdings vorher die Fähigkeit an unzugänglichen Stellen landen zu können. Diese magischen Flügel gestatten es, Bilder so aufzubauen, daß nur der taktisch gezielte Einsatz dieser Flughilfe es ermöglicht, ein Bild vollständig abzuräumen. Dabei kann es vorkommen, daß sich der Vogel in eine völlig aussichtslose Situation manövriert, und in einer Sackgasse landet. Auch hier bietet das Programm einen Ausweg an, die Funktionstasten. Diese können jederzeit (außer während der Flugphase) benutzt werden.

Bei dem ersten Start des Programms dauert es einige Zeit (bei weiteren Starts geht es dann schneller), bis sich das Programm mit der Frage »Edit, Wahl oder Spiel?« meldet.

Mit »W« kann man eines der sechs Bilder zum Spielen auswählen, mit »S« wird ein vollständiges Spiel ab Bild 1 gewählt.

Die Funktionstasten haben folgende Bedeutung:

f1: Dieses Spiel aufgeben, zurück zum Menü
f2: diesen Vogel opfern und mit dem nächsten Vogel neu an der Startposition beginnen
f3: mit dem nächsten Bild weitermachen
f5: Pause (weiter mit SPACE)

Für jedes Ei gibt es je nach Bild 10, 20 ... 60 Punkte. Wird ein Bild in einer bestimmten Zeit beendet, so erhält der Spieler einen Bonus, der um so größer ist, je schneller das Bild beendet wurde.

Der Editor

Der Editor dient dazu, eigene Spiele zu entwickeln. Mit ihm können neue Bilder erstellt werden, die im Programmtext selbst die alten Bilder überschreiben. Deshalb sollte das Programm vor dem Editieren, das neue Spiel nach dem Editieren unter neuem Namen gespeichert werden. Dazu einfach das Programm mit »STOP« abbrechen. Der Editor fragt zunächst, welches Bild geändert werden soll. Dann erscheint dieses Bild mit einem Cursor. Es stehen folgende Funktionen zur Verfügung:

←: die vorgenommenen Änderungen endgültig in den Programmtext übernehmen
f1: Editor verlassen, ohne Änderungen zu übernehmen. Das Bild bleibt erhalten, wie es vor dem Editoraufruf war. In beiden Fällen fragt das Programm, ob das Bild ausprobiert werden soll. Aus dem Spiel gelangt man mit f1 und E jederzeit in den Editor zurück.

Die Cursorsteuerung erfolgt wie gewohnt über die Cursortasten. SHIFT CLR: Bild löschen

Die Startposition des Vogels wird durch »0« festgelegt. In jedes Bild können zwischen 0 und 7 Monster gesetzt werden, die sich horizontal auf einer Breite von acht Spalten hin und her bewegen. Das linke Ende dieser Bewegungsbahn kann mit den Tasten 1 bis 7 festgelegt werden.

Mit den Tasten A bis T können die 20 verschiedenen Bausteine, die zur Verfügung stehen, gesetzt werden. Der Vogel reagiert immer nur auf das Zeichen unter seinen Füßen. Bei Sprüngen reagiert er (außer bei Wänden) erst in der absteigenden Phase wieder auf Zeichen. Eine Übersicht über die Bausteine bietet die Tabelle 1. Dazu einige ergänzende Bemerkungen:

Die Bausteine A bis D sind durch verschiedene Bewegungsmöglichkeiten gekennzeichnet. Springen ist möglich:

A: rechts, links
B: rechts, links, runter
C: rechts, links, rauf
D: alle vier Richtungen

Bei E bis H ist eine Bewegung nur in jeweils eine Richtung und kein Sprung möglich.

Bei I bis N wird der Vogel automatisch bewegt. Er selbst kann nur springen.

Der Trampolin O bewirkt einen zufälligen Sprung nach rechts oder links.

Bei der Gummiwand S wird der Vogel auf das letzte Zeichen, das weder Wand noch Luft war, zurückgeworfen.

Da das Problem in die Interruptroutine eingreift und seinen eigenen Programmtext verändert, sollte es auf jeden Fall vor dem ersten Start abgespeichert werden.

Um Tippfehler bei den zahlreichen DATA-Zeilen besser lokalisieren zu können, wurden diese in verschiedene Blöcke mit jeweils eigener Prüfsumme aufgeteilt.

Bitte halten Sie sich beim Eintippen der sechs Bilder exakt an die vorgegebene Zeilenlänge: Jeder PRINT-Befehl muß genau 39 Zeichen lang sein. Fehlende Zeichen können zur Zerstörung des Programms führen, da der Editor diese Zeilen überschreibt.

Hinweise zum Eintippen

Daß die jeweils 39 Zeichen genau mit dem Listing übereinstimmen, ist dagegen nicht so wichtig. Durch Abweichungen wird hier allenfalls die Spielbarkeit des vorgegebenen Spiels beeinflußt, ein Mangel, der jederzeit mit dem Editor behoben werden kann. Im Prinzip ist es auch möglich, auf das Abtippen der Bilder 2 bis 6 zu verzichten und an anderer Stelle einfach mit Hilfe des Bildschirmeditors fünfmal das erste Bild zu kopieren. Auf jeden Fall aber muß die Zahl der Zeilen, die Zeilennummer und die Länge der Zeilen mit dem Listing übereinstimmen.

Um das Eintippen der Bilder zu erleichtern, wurden diese in der vorliegenden Programmversion in den Programmtext selbst gelegt. Deshalb ist die Zahl der Bilder auf sechs (diese Zahl wurde gewählt, um die Tipparbeit in erträglichen Grenzen zu halten) festgelegt. Die Anzahl der Bilder kann aber folgendermaßen erhöht werden:

Der Variablen BM in Zeile 50070 ist die gewünschte Anzahl zuzuweisen. Für jedes weitere Bild sind dem Programm entsprechend dem Schema der ersten sechs Bilder 22 PRINT-Zeilen, die jeweils 39 beliebige Zeichen ausgeben, und ein abschließendes RETURN, anzufügen. Die erste Zeilennummer eines Bildes berechnet sich nach der Formel 59900 + Bildnummer * 100 +1, zum Beispiel 60601 für Bild 7.

(Matthias Törk/og)

TASTE	: BAUSTEIN
A	: BODEN
B	: LEITER UNTERES ENDE
C	: LEITER OBERES ENDE
D	: LEITER MITTELSTÜCK
E	: SEIL HOCH
F	: SEIL RUNTER
G	: EINBAHNSTRASSE RECHTS
H	: EINBAHNSTRASSE LINKS
I	: TRANSPORTBAND RECHTS
J	: TRANSPORTBAND LINKS
K	: AUFZUG HOCH
L	: AUFZUG RUNTER
M	: RUTSCHBAHN RECHTS RUNTER
N	: RUTSCHBAHN LINKS RUNTER
O	: TRAMPOLIN
P	: LUFT, FREIER FALL
Q	: TOEDLICHE FALLE
R	: EI
S	: GUMMI-WAND
T	: MAGISCHE FLÜGEL

Tabelle 1. Die verschiedenen Bausteine und ihre Bedeutung

Die wichtigsten Routinen

50000	:INITIALISIERUNG
48000	:MENUE
49000	:EDITOR
45000	:AB BILD BI SPIELEN
1050	:EIN BILD SPIELEN
46000	:BILD BI INITIALISIEREN
500	:JOYSTICK
600	:FIGUREN BEWEGEN
40000	:SPRINGEN
11000	:EI
2000	:LUFT
12000	:TOEDLICHE FALLE
15000	:FLIEGEN

Tabelle 3. Die wichtigsten Routinen

Die wichtigsten Variablen

XS,YS	: Spritekoordinaten des Helden	CO ( )	: Farbtabelle
XG,YG	: Koordinatengrenzen	JP,JS	: Joystick
OX,OY	: Koordinatenoffset Sprite/Zeichen	EZ,EM	: Anzahl Eier
TG	: Zeitlimit	SC	: Punkte
BM	: maximale Bildzahl	MZ	: Anzahl Leben
BI	: aktuelles Bild	CP	: Cursorposition
UP ( ),		BS	: Bildschirmadresse
DO ( ),RI ( ),		FR	: Adresse Farb-RAM
LE ( ),DX ( ),		KO	: Sprite-Kollisionsregister
DY ( )	: Bewegungstabellen	V	: VIC-Chip
JU ( )	: Sprungtabelle	ZS	: Adresse Zeichensatz
		EA	: Adresse Bilder
		SI,W1,GL,GH	: Sound

Tabelle 2. Die wichtigsten Variablen von Spring-Vogel



```
10 REM *** JUMP.80/  8.6.84                      <066>
20 POKE 53280,0:POKE 53281,0                     <148>
25 PRINT"{CLR}NICHT WUNDERN UND WARTEN!!"        <160>
30 GOSUB 50000:GOTO 48000                        <029>
499 REM ** JOYSTICK **                           <171>
500 JS=PEEK(JP):IF (JS AND 16)=0 THEN GOSU
    B 40000:GOTO 600                             <139>
505 IF (JS AND Q1)=Q0 THEN POKE W1,SW:IF U
    P(CM) THEN YS=YS-Q8:GOTO 590                 <030>
510 IF(JS AND Q2)=Q0 THEN POKE W1,SW:IF DO
    (CM) THEN YS=YS+Q8:GOTO 590                  <091>
520 IF(JS AND Q4)=Q0 THEN POKE W1,SW:IF LE
    (CM) THEN XS=XS-Q4:GOTO 590                  <009>
530 IF(JS AND Q8)=Q0 THEN POKE W1,SW:IF RI
    (CM) THEN XS=XS+Q4                           <186>
590 POKE W1,Q0                                   <110>
600 IF XS<OX THEN XS=OX                          <046>
620 SYS(SR) Q0,XS,YS:RETURN                      <139>
1000 REM *** MAIN LOOP **                        <132>
1050 IF PEEK(TA)THEN 1200                        <171>
1055 IF PEEK(KO)AND Q1 THEN CM=16:POKE KO,
     0:GOTO 1100                                 <103>
1060 CM=USR(XS-OX) ,YS-OY:IF CM=Z8 THEN XS
     =8*PEEK(49426)+OX:YS=8*PEEK(49427)+OY <234>
1070 IF CM>Q7 THEN XS=XS+DX(CM):YS=YS+DY(C
     M)                                          <033>
1075 IF TI<TG THEN PRINT"{HOME}",INT(TI/10
     0)                                          <103>
1080 IF CM<Z4 THEN GOSUB 500:GOTO 1050           <055>
1100 ON CM-13 GOTO 3000,2000,12000,11000,1
     050,15000                                   <155>
1200 GET B$:IF B$="{F1}"THEN RETURN              <210>
1210 IF B$="{F3}" THEN CM=16:GOTO 1100           <123>
1215 IF B$="{F5}" THEN 11040                     <061>
1217 IF B$="{F7}" THEN POKE 198,0:WAIT 198
     ,1                                          <194>
1220 GOTO 1050                                   <048>
2000 I=0:POKE W1,33                              <096>
2010 I=I+1:YS=YS+4:IF I>8 OR YS>YG THEN CM
     =16:POKE W1,0:POKE SI+1,GH:GOTO 1100        <133>
2015 POKE SI+1,40-2*I                            <246>
2020 GOSUB 600:CM=USR(XS-OX) ,YS-OY:IF CM=
     15 THEN   2010                              <191>
2025 IF CM=18 THEN XS=FN X(XS)                   <074>
2030 POKE W1,0:POKE SI+1,GH:YS=FN Y(YS):GO
     TO 1050                                     <121>
3000 RX=2-4*INT(RND(1)*2):GOSUB 40100:GOSU
     B 600:GOTO 1050                             <123>
10999 REM ** GEGENSTAND POSITIV, EI**            <074>
11000 POKE BS+40*PEEK(49427)+PEEK(49426),8
      0:SC=SC+BI*10:EZ=EZ+1                      <115>
11030 IF EZ<EM THEN PRINT"{HOME}",,SC:GOTO
       1050                                      <070>
11035 IF TI<TG THEN SC=SC+INT((TG-TI)/10)        <088>
11040 BI=BI+1:IF BI>BM THEN RETURN               <179>
11050 GOSUB 46000:GOTO 1050                      <207>
11999 :                                          <035>
12000 MZ=MZ-1:IF MZ=0 THEN RETURN                <039>
12050 PRINT"{HOME}",,SC,MID$("******{6SPAC
      E}",7-MZ,6)                                <117>
12052 POKE W1,17                                 <009>
12055 FOR XS=XS TO XX STEP SGN(XX-XS):GOSU
      B 600:POKE SI+1,49+(XS AND 1):NEXT         <150>
12057 FOR YS=YS TO YY STEP SGN(YY-YS):GOSU
      B 600:POKE SI+1,52+(YS AND 1):NEXT         <171>
12060 XS=XX:YS=YY:POKE KO,0:POKE W1,0            <004>
12070 GOTO 1050                                  <232>
15000 POKE V+29,255:POKE BS+40*PEEK(49427)
      +PEEK(49426),80:CM=3:XS=XS-10:SW=17        <142>
15050 JS=PEEK(56320)                             <159>
15052 IF(JS AND 16)=0 THEN XS=XS+10:POKE V
      +29,254:GOSUB 600:POKE KO,0:SW=129:G
      OTO 1050                                   <045>
15055 IF YS<OY THEN YS=OY                        <191>
15056 IF YS>YG THEN YS=YG                        <088>
15057 IF XS>XG THEN XS=XG                        <255>
15060 GOSUB 505:GOTO 15050                       <041>
39000 :                                          <113>
39999 REM **** JUMP ****                         <143>
40000 IF CM>3 AND CM<8 THEN RETURN               <037>
40005 RX=0:IF (JS AND 4)=0 THEN RX=-2            <000>
40010 IF (JS AND 8)=0 THEN RX=2                  <135>
40100 POKE SI+6,16*15+13:POKE W1,17:POKE W
      1,16:FOR I=0 TO 13                         <132>
40120 POKE SI+1,104-JU(I):YS=YS-JU(I):XS=X
      S+RX:GOSUB 600                             <221>
40130 CM=USR(XS-OX) ,YS-OY:                      <007>
40140 IF CM=18 THEN XS=FN X(XS):GOTO 40300 <233>
40150 IF I<9 THEN POKE KO,0:GOTO 40250           <012>
40235 IF CM<>15 OR (PEEK(KO)AND 1) THEN 40
      300                                        <233>
40250 NEXT                                       <127>
40300 IF CM<>15 THEN YS=FN Y(YS)                 <172>
40310 POKE W1,0:POKE SI+6,16*15:POKE SI+1,
      GH:RETURN                                  <041>
44990 :                                          <005>
45000 MZ=6:SC=0:GOSUB 46000                      <034>
45100 GOSUB 1050:POKE V+21,0                     <064>
45150 FOR I=1 TO 3000:NEXT                       <107>
45160 PRINT"{CLR,4DOWN,YELLOW}","{GREEN}DA
      S WARS!!!":PRINT"{DOWN}ERREICHTE PUN
      KTZAHL:{RVSON}",SC                         <121>
45163 IF SC>HS THEN HS=SC                        <111>
45165 PRINT"{DOWN}HIGH SCORE:{RVSON}",,HS        <206>
45170 PRINT"{RED}--------------------------
      -------------{LIG.BLUE}":PRINT,"{DOWN
      }** JUMP!! **"                             <164>
45180 IF (PEEK(56320) AND 16)<>0 THEN 4518
      0                                          <090>
45190 RETURN                                     <033>
46000 SYS(49155) 0:EZ=0:XX=120:YY=120            <134>
46010 POKE W1,0:POKE SI,GL:POKE SI+1,GH:SW
      =129:REM SOUND                             <243>
46015 A=0:IF TI<TG THEN A=INT((TG-TI)/10)        <190>
46020 PRINT"{CLR,YELLOW,3SPACE}BONUS:",A,"
      {4LEFT}PKT:";SC,MID$("******{6SPACE}
      ",7-MZ,6):PRINT                            <132>
46030 FOR I=1 TO 7:POKE XH+I,1:POKE XL+I,1
      70:POKE TY+I,0:NEXT                        <063>
46050 ON BI GOSUB 60000,60100,60200,60300,
      60400,60500                                <248>
46100 EM=0:FOR I=80 TO 958:C=PEEK(BS+I):IF
       C=80 THEN NEXT:GOTO 46200                 <067>
46105 IF C=82 THEN EM=EM+1                       <163>
46110 IF C>64 AND C<85 THEN POKE FR+I,CO(C
      -65):NEXT:GOTO 46200                       <108>
46120 POKE FR+I,0:C=C-48:XS=(I-INT(I/40)*4
      0)*8+OX:YS=INT(I/40)*8+OY                  <019>
46121 IF C<0 OR C>7 THEN 46140                   <244>
46130 POKE TY+C,YS+12:POKE XH+C,XS/256:POK
      E XL+C,XS AND 255:IF C=0 THEN XX=XS:
      YY=YS                                      <247>
46140 NEXT                                       <175>
46200 TI$="000000":TG=5000+2000*BI:YS=YY:X
      S=XX                                       <087>
46220 PRINT"{HOME}B";BI;"{LEFT,2SPACE}ZEIT
      :";"{6SPACE}":RETURN                       <088>
48000 REM *** MENUE ***                          <167>
48040 POKE V+21,0:PRINT"{CLR,LIG.BLUE,DOWN
      }":GOSUB 60500                             <144>
48041 PRINT"{5UP}","{ORANGE,RIGHT}SF1: AUF
      GEBENS":PRINT,"{RIGHT}SF3: ← VOGEL S
      "                                          <189>
48042 PRINT,"{RIGHT}SF5: ← BILD{2SPACE}S":
      PRINT,"{RIGHT}SF7: PAUSE{3SPACE}S"         <093>
48043 PRINT"{HOME,YELLOW}AAAAAAAAAAAA SPRI
      NG VOGEL AAAAAAAAAAAA"                     <010>
48044 PRINT"{GREEN,RVSON}E{RVOFF}DIT,{SPAC
      E,RVSON}W{RVOFF}AHL ODER{SPACE,RVSON
      }S{RVOFF}PIEL";                            <143>
48045 INPUT B$:IF B$="E"THEN GOSUB 49000:G
      OTO 48040                                  <213>
48050 IF B$="S" THEN BI=1: GOSUB 45000:GOT
      O 48040                                    <041>
48060 IF B$<>"W" THEN 48040                      <048>
48070 INPUT"{CLR,DOWN}WELCHES BILD SPIELEN
      ";B$:BI=VAL(B$):IF BI<1 OR BI>BM THE
      N 48070                                    <122>
48080 GOSUB 45000:GOTO 48040                     <014>
48090 :                                          <057>
48999 REM *** EDITOR ****                        <021>
49000 INPUT"{CLR,3DOWN}WELCHES BILD";B$:BI
      =VAL(B$):IF BI<0 OR BI>BM THEN 49000 <032>
49020 PRINT"BITTE WARTEN":ZN=59900+BI*100:
      ZA=EA                                      <165>
49050 IF FN AD(ZA+2)<ZN THEN ZA=FN AD(ZA):
      GOTO 49050                                 <135>
49055 POKE 53281,1:PRINT"{CLR,DOWN}":POKE
      53281,0                                    <001>
49060 ON BI GOSUB 60000,60100,60200,60300,
      60400,60500                                <210>
49069 REM FARBE SETZEN                           <073>
```




```
49070 FOR I=80 TO 958                                  <144>
49071 C=PEEK(BS+I)-65:IF C>=0 AND C<20 THE
      N POKE 55296+I,CO(C):NEXT:GOTO 49095 <208>
49075 POKE 55296+I,1:NEXT                              <163>
49095 CP=80:PRINT"{HOME,LIG.BLUE}ZEICHEN:{
      SPACE,RVSON}A-T{RVOFF,2SPACE}POS:{SP
      ACE,RVSON}0-7{RVOFF,2SPACE,RVSON}CUR
      S.{RVOFF,2SPACE,RVSON}CLR{RVOFF}"                <003>
49096 PRINT"AENDERUNGEN UEBERNEHMEN:{SPACE
      ,RVSON}←{RVOFF,2SPACE}QUIT:{SPACE,RV
      SON}F1{RVOFF}"                                   <050>
49100 POKE 646,PEEK(CP+55296):POKE 204,0               <220>
49105 GET B$:IF B$="" THEN 49105                       <022>
49110 POKE 207,0:POKE 204,1:POKE BS+CP,PEE
      K(BS+CP)AND 127                                  <184>
49120 C=ASC(B$):IF C>64 AND C<85 THEN POKE
       646,CO(C-65):PRINT CHR$(C+32);:GOTO
       49140                                           <211>
49125 IF B$>="0"AND B$<"8"THEN PRINT"{WHIT
      E}";B$;                                          <126>
49130 IF B$="{DOWN}" OR B$="{UP}" OR B$="{
      RIGHT}" OR B$="{LEFT}" THEN PRINT B$
                                                       <141>
49140 IF PEEK(211)>38 THEN PRINT"{LEFT}";              <023>
49150 CP=PEEK(211)+40*PEEK(214):IF CP>=24*
      40 OR CP<80 THEN CP=80:PRINT"{HOME,D
      OWN}"                                            <228>
49155 IF B$="{CLR}"THEN POKE 53281,1:PRINT
      "{CLR,DOWN}":POKE 53281,0:GOTO 49095 <189>
49160 IF B$<>"←" AND B$<>"{F1}" THEN 49100 <139>
49170 IF B$="{F1}" THEN RETURN                         <096>
49200 B=BS+40:PRINT"{HOME,6SPACE}**** BITT
      E WARTEN ***{8SPACE}"                            <171>
49210 FOR I=0 TO 21:ZA=FN AD(ZA):A=ZA+6:B=
      B+40                                             <231>
49220 FOR J=0 TO 38:C=PEEK(B+J)                        <016>
49221 IF C>=48 AND C<=55 THEN C=C-32:GOTO
      49230                                            <018>
49229 IF C<65 OR C>84 THEN C=80                        <189>
49230 POKE A+J,C+32:NEXT J,I                           <159>
49350 INPUT"{CLR,DOWN}BILD SPIELEN (J/N)";
      B$:IF B$="N" THEN RETURN                         <214>
49370 GOSUB 45000:RETURN                               <006>
49499 RETURN                                           <024>
49999 REM                                              <020>
50000 REM ** INIT **                                   <184>
50010 CM=0:XS=0:YS=0:JS=0:OX=14:OY=36:T=0:
      TG=2000                                          <077>
50020 Q0=0:Q8=8:Q4=4:Q1=1:Q2=2:Q7=7:Z8=18:
      Z4=14:JP=56320:SR=49249:TA=198                   <116>
50040 XH=49400:XL=49392:TY=49432                       <042>
50042 DEF FN Y(Y)=PEEK(49427)*8+OY                     <154>
50045 DEF FN X(X)=PEEK(49426)*8+OX                     <173>
50050 V=53248:BS=256*204:XG=320+OX:YG=200+
      OY                                               <120>
50060 KO=V+30:FR=55296:REM SPR.KOLLIS.REG
      / FARBRAM                                        <076>
50070 POKE 786,193:POKE 785,32:BM=6: REM U
      SR ADR.,MAX.BILDER                               <074>
50090 DIM JU(13):FOR I=0 TO 13:READ JU(I):
      NEXT                                             <081>
50100 DATA 5,4,3,2,1,1,0,0,-1,-1,-2,-3,-4,
      -5                                               <067>
50215 FOR I=BS+1017 TO BS+1023:POKE I,16:N
      EXT:POKE BS+1016,14                              <094>
50220 POKE V+23,0:POKE V+29,254:POKE V+39,
      7                                                <207>
50230 DIM X(7),Y(7),RX(7),RY(1):FOR I=0 TO
       7:X(I)=100:Y(I)=100:NEXT                        <127>
50249 REM MAPRO-TABELLEN                               <159>
50250 FOR I=1 TO 7:POKE V+39+I,I:POKE 4943
      2+I,30*I:POKE 49392+I,35*I:POKE 4940
      0+I,0                                            <004>
50251 POKE 49416+I,1:POKE 49408+I,5+RND(1)
      *45:NEXT:POKE 49401,1                            <156>
50252 POKE 49424,1:REM SB-FLAG                         <091>
50410 DIM UP(20),DO(20),RI(20),LE(20),DX(2
      0),DY(20)                                        <206>
50422 RI(0)=1:LE(0)=1:RI(1)=1:LE(1)=1:UP(1
      )=1:RI(2)=1:LE(2)=1:DO(2)=1                      <215>
50437 UP(3)=1:DO(3)=1:RI(3)=1:LE(3)=1:UP(4
      )=1:DO(5)=1:RI(6)=1:LE(7)=1:DX(8)=4  <035>
50477 DX(9)=-4:DY(10)=-8:DY(11)=8:DY(12)=8
      :DX(12)=8:DY(13)=8:DX(13)=-8:DY(15)=
      4                                                <038>
50599 REM SOUND                                        <093>
50600 SI=54272:W1=SI+4                                 <241>
50620 POKE SI+24,15                                    <048>
50625 POKE SI+5,0:POKE SI+6,15*16+0                    <082>
50630 GL=180:GH=18:A=0                                 <031>
50710 FOR I=13 TO 16:FOR N=0 TO 62:READ Q:
      A=A+Q:POKE 49152+I*64+N,Q:NEXT:NEXT              <041>
50720 IF A<>16572 THEN PRINT"DATAERROR":ST
      OP                                               <041>
50800 DATA 0,255,0,1,255,128,1,153,128,1,1
      53,128,3,255,192,31,195,248                      <075>
50805 DATA 63,227,252,103,243,230,195,249,
      195,131,253,193,3,255,192                        <082>
50807 DATA 3,255,192,1,231,128,0,231,0,0,1
      02,0,0,102,0,0,102,0,1,231,128                   <231>
50809 DATA ,,,,,,,,                                    <017>
50810 DATA 0,255,0,1,255,128,129,153,129,1
      93,153,131,99,255,198,63,195,252                 <110>
50815 DATA 31,199,248,7,207,224,3,159,192,
      3,191,192,3,255,192                              <165>
50817 DATA 3,255,192,1,231,128,0,231,0,0,1
      02,0,0,102,0,0,102,0,1,231,128                   <241>
50819 DATA ,,,,,,,,                                    <027>
50850 DATA 0,192,0, 1,224,0, 3,240,0, 2,20
      8,0, 2,208,0,3,240,0,3,240,0                     <012>
50855 DATA 3,48,0,15,112,0                             <021>
50857 DATA ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
      ,,,,                                             <224>
50870 DATA 0,192,0,1,224,0,3,240,0, 2,208,
      0, 2,208,0, 3,240,0, 3,240,0                     <032>
50875 DATA 3,48,0,3,188,0                              <126>
50880 DATA ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
      ,,,,                                             <247>
51000 DIM CO(19):FOR I=0 TO 19:READ CO(I):
      NEXT                                             <190>
51010 DATA 5,5,5,2,10,8,13,13,5,5,4,4,4,4,
      12,1,2,7,8,14                                    <139>
52000 REM EDIT-ANFANGSADR.                             <121>
52010 DEF FN AD(X)=PEEK(X)+256*PEEK(X+1)               <194>
52020 EA=2049                                          <186>
52050 IF FN AD(EA+2)<59800 THEN EA=FN AD(E
      A+2):GOTO 52050                                  <184>
58000 REM ZEICHENSATZ NACH $E0000 (ZS), VI
      DERRAM NACH $CC00 (BS)                           <087>
58020 ZS=14*4096:ZV=53248                              <172>
58040 IF(PEEK(53272)AND 254)=56 THEN RETUR
      N                                                <185>
58050 POKE 53272,56:POKE 56576,148:POKE 64
      8,204:BS=204*256                                 <212>
58055 PRINT"{CLR,7SPACE}GEDULD":PRINT:GOSU
      B 60400                                          <186>
58060 POKE 56334,0:POKE 1,51                           <244>
58070 FOR I=0 TO 2047:POKE ZS+I,PEEK(ZV+I)
      :NEXT                                            <225>
58075 POKE 1,55:POKE 56334,1:A=0                       <046>
58090 READ C:A=A+C:IF C>255 THEN 59010                 <088>
58095 H=ZS+8*(C+65):FOR I=0 TO 7:READ C:A=
      A+C:POKE H+I,C:NEXT                              <167>
58098 GOTO 58090                                       <117>
58100 DATA 0,255,60,24,24,24,60,255,255                <067>
58110 DATA 1,66,66,66,66,255,24,24,255                 <036>
58120 DATA 2,255,24,24,255,66,66,66,66                 <117>
58130 DATA 3,66,66,126,66,66,66,126,66                 <206>
58140 DATA 4,12,24,48,0,12,24,48,0                     <132>
58150 DATA 5,48,24,12,0,48,24,12,0                     <075>
58160 DATA 6,255,129,8,4,62,4,8,255                    <207>
58170 DATA 7,255,129,16,32,124,32,16,255               <108>
58180 DATA 8,255,56,28,14,14,28,56,255                 <093>
58190 DATA 9,255,28,56,112,112,56,28,255               <058>
58200 DATA 10,66,66,90,102,102,66,66,66                <091>
58210 DATA 11,66,66,102,102,90,66,66,66                <049>
58220 DATA 12,192,208,112,56,28,14,7,3                 <126>
58230 DATA 13,3,7,14,28,56,112,208,192                 <151>
58240 DATA 14,255,62,12,48,192,48,12,62                <032>
58250 DATA 15,0,0,0,0,0,0,0,0                          <225>
58260 DATA 16,0,126,126,102,102,126,126,0              <053>
58270 DATA 17,24,60,60,126,126,126,60,24               <239>
58280 DATA 18,0,223,223,0,0,251,251,0                  <214>
58290 DATA 19,0,102,255,153,153,56,56,0                <129>
58299 DATA 999                                         <019>
59000 REM ** MAPROS **                                 <199>
59010 IF A<>14182 THEN PRINT"DATA-ERROR IN
       58100FF":STOP                                   <105>
```

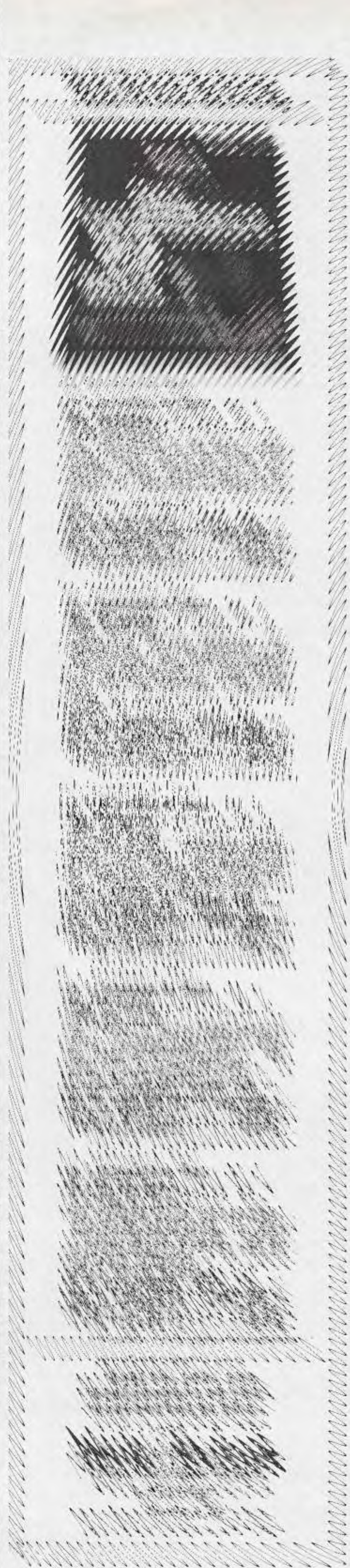
Listing »Springvogel«.
Bitte beachten Sie die Eingabehinweise auf Seite 16.

```
59014 A=0:FOR I=49152 TO 49378:READ N:POKE
       I,N:A=A+N:NEXT I                       <085>
59016 IF A<>27365 THEN PRINT"DATA ERROR":S
      TOP                                     <238>
59020 DATA 80,70,83,32,158,183,224,8,176,3
      1,189,219,192,45,21,208,141             <077>
59022 DATA 21,208,96,32,158,183,224,16,176
      ,14,134,2,32,253,174,32,235             <202>
59024 DATA 183,165,21,201,2,144,3,76,72,17
      8,138,72,166,2,32,10,192                <031>
59026 DATA 189,219,192,45,16,208,141,16,20
      8,70,21,144,9,189,211,192,13            <162>
59028 DATA 16,208,141,16,208,138,10,170,10
      4,157,1,208,165,20,157,0,208            <063>
59030 DATA 166,2,189,211,192,13,21,208,141
      ,21,208,96                              <241>
59099 REM                                     <230>
59100 DATA 32, 20,192,173, 16,193, 73,254,
      141, 16                                 <092>
59102 DATA 193,162, 7,189,248,207, 24,109,
       16,193                                 <080>
59104 DATA 157,248,207,202, 16,243,169, 7,
      133, 2                                  <208>
59106 DATA 166, 2,189, 0,193, 24,125, 8,19
      3,157                                   <163>
59108 DATA 0,193,201, 64,144, 11,189, 8,19
      3, 73                                   <137>
59110 DATA 254,157, 8,193,176,232,234, 24,
      125,240                                 <096>
59112 DATA 192,133, 20, 42, 29,248,192, 41
      , 1,234                                 <199>
59114 DATA 234,133, 21,189, 24,193, 32, 45
      ,192,198                                <012>
59116 DATA 2,208,203,173,30,208,96            <100>
59140 DATA 253,174,32                         <217>
59142 DATA 9,185,211,192,13,29,208,141,29,
      208,70,2,144,9,185,211,192              <253>
59144 DATA 13,23,208,141,23,208,96,1,2,4,8
      ,16,32,64,128,254,253                   <072>
59146 DATA 251,247,239,223,191,127            <067>
59190 REM                                     <067>
59195 A=0:FOR I=49440 TO 49505:READ N:A=A+
      N:POKE I,N:NEXT                         <228>
59196 IF A<>8224 THEN PRINT"DATA ERROR":ST
      OP                                      <028>
59200 DATA 32,238,183,102, 21,165, 20,106,
       74, 74                                 <086>
59202 DATA 133,211,201, 40,144, 4,169, 15,
      176, 42                                 <180>
59204 DATA 133,211,138, 74, 74, 74,133,214
      , 32,108                                <241>
59206 DATA 229,164,211,177,209, 56,233, 65
      ,201, 20                                <204>
59208 DATA 144, 2,169, 15,201, 15,240, 14,
      201, 18                                 <240>
59210 DATA 240, 10,166,211,142, 18,193,166
      ,214,142                                <073>
59212 DATA 19,193,168, 76,162,179             <234>
59800 RETURN                                  <165>
59999 REM BILD 1                              <074>
60000 :                                       <029>
60001 PRINT"PPPPPPPPPPPPPRPPPPPPPPPPPPPPPPP
      PPPPPPPPS"                              <060>
60002 PRINT"PPPPLJJJJAAAAAAAAQAAAAPPPIIIIT
      PPPPPPRPS"                              <003>
60003 PRINT"PPPPLPPPPPEPPPPPPPPPPPPPPPPPPPP
      PPPPPPPPS"                              <115>
60004 PRINT"PPPPLPPPPPEPPPPPPPPPPPPPPPPPPPP
      PRPPAAAAS"                              <011>
60005 PRINT"PPPPLPPPPPEPPPPPPPPPPPPPPPPPPPA
      AAAAPPPSS"                              <214>
60006 PRINT"PPRPLPRPPPEPPPPPPPPPSSSSSSSSSPP
      PPPPPPPSS"                              <131>
60007 PRINT"PAAACAAAPPEAAAAPPPPPPPPPPPPPPPP
      PPPPPPRSS"                              <133>
60008 PRINT"PPPPFPPPPPEPPPPPPP2PPPPPPPPPPPP
      PPPPPPPSS"                              <232>
60009 PRINT"PPPPFPPPPPEPPPPAAAAAAAAAAAAAAAA
      AOAAAAASS"                              <067>
60010 PRINT"PPPPFPPPPPEPPPRPPPPPPPPPPPPPPPP
      PPPPPPDPP"                              <100>
60011 PRINT"PPPPFPPPPPEPPAAAAAPPPPPPPPPPPPP
      PPPPPPDPP"                              <184>
60012 PRINT"PPPPFPPPPPEPPPPPPPPPPPPPPPPPPPP
      PPPPPPDPP"                              <094>
60013 PRINT"PPP1FPPPPPEPPPPPPPPPPPPPPPPPPPP
      PPPPPPDPP"                              <191>
60014 PRINT"PPPPFPPPPPEPPPPPPPPPPPPPPPPPPRH
      HHHHHABAC"                              <146>
60015 PRINT"PPPPPPPPPPEPPPPPPPPPRPPPPPPP3PFP
      PPPPPPPPD"                              <238>
60016 PRINT"PPPPRPPPPPEPPPPPPPPPAPPPPPPPPPFP
      PPPPPPPPD"                              <184>
60017 PRINT"PPPPGGGGGGEPPPPPPPRKRPPPPPPPPFP
      PPPPPPPPD"                              <004>
60018 PRINT"PPPPPPPPPPPPPPPPPPPAKAPPPIIPPFP
      PPPPPPRPD"                              <093>
60019 PRINT"PPPPPPPPPPPPPPPPPPPPKPPPPKPRPFP
      PPPRPPAAB"                              <187>
60020 PRINT"PPPPPPPPPPRPPPPPPPPPKPPPPKPPPPP
      PAAAAPPPP"                              <220>
60021 PRINT"PØPPRPPPPAAAAIIIRPPKPPPPKAAAAP
      PPPPPPPPP"                              <170>
60022 PRINT"AAAAAAAAASSSSSSSAAABAIIIKSSSSS
      SSSSSSSSS"                              <114>
60030 RETURN                                  <141>
60100 :                                       <129>
60101 PRINT"PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
      PPPPPPPPP"                              <144>
60102 PRINT"PPPPPRPPRPPPPPPPPPPPPPPPPPPPPRP
      PPRPPPPPP"                              <215>
60103 PRINT"AARAAAAAAAAAPPPAAAARPRAAAAPPAAAA
      AAAAAAARAA"                             <206>
60104 PRINT"PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
      PPPPPPPPP"                              <147>
60105 PRINT"PPRPPPPPIIIIP4PPPPPPPPPPPPJJJJP
      PPPPPPRPP"                              <186>
60106 PRINT"PPPPPPPPPPPPPRJJJJAIIIIRPPPPPP
      PPPPPPPPP"                              <198>
60107 PRINT"PPRIIIIPPPPPPPPPPPPPPPPPPPPPPPP
      JJJJJPRPP"                              <119>
60108 PRINT"PPPPPPPJJJPPPPPPPPPPPPPPPPPPIII
      PPPPPPPPP"                              <172>
60109 PRINT"PPRPPPPPPPPJJJJPPPPPPPPPPPIIIIPPP
      PPPPPPRPP"                              <055>
60110 PRINT"PPPPPPPPPPPPPPJJJPPPIIIPPPPPPP
      PPPPPPPPP"                              <191>
60111 PRINT"PPØPPPPPPPPPPPPPPPPPPPPPPPPPPPP
      PPPPPPPPP"                              <138>
60112 PRINT"CAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAAA"                              <051>
60113 PRINT"DSPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
      PPPPPPPPP"                              <059>
60114 PRINT"DSPRPPPPPPPPPPPPPPPPPPPPPPPPPPP
      PPPPPPPTP"                              <069>
60115 PRINT"DSPAPPPPPPPPPPPPPPPPPPPPPPPPPPP
      PPPPPPPAP"                              <151>
60116 PRINT"DSPPP5PPPPPRPPPPPRPPPPPPPRPP6PP
      RPPPPPPPR"                              <028>
60117 PRINT"DSAACPPPPCAAAAAAAAAAAAAAAAAAAA
      AAAACAAAA"                              <015>
60118 PRINT"DPPPDPPPPDPPPPPPPPPPPPPPPPPPPPP
      PPPPDPPPP"                              <079>
60119 PRINT"DTPPDPPPPDPPPPPPPPPPPPPPPPPPPPP
      PPPPDPPPP"                              <208>
60120 PRINT"DAPPDPPPPDPPPPRPPRPPPPRPPRPPRPP
      RPPPDPPPR"                              <236>
60121 PRINT"DPPPDPPP2DPPPPPPPPPPPPPP3PPPPPP
      PPPPDPPPP"                              <132>
60122 PRINT"BAAABAAAABAAAAAAAAAAAAAAAAAAAAA
      AAAABAAAA"                              <142>
60150 RETURN                                  <005>
60199 REM BILD3                               <021>
60200 :                                       <231>
60210 PRINT"SSSSRHHPQPJJJCIIIPPPIIIRIIPPPA
      AAAAAAAAR"                              <191>
60214 PRINT"PPPSFSPPPPSPPDPSSPPQPPPLPPPPPP
      PPPPPPPPN"                              <200>
60216 PRINT"IIRSFSPPPPSPPDPSSPPPPSPLPPPPPP
      PPØPPPPNP"                              <126>
60218 PRINT"KPPSFSPPRPSPPDPSSPPPRSPLPPPPPP
      PPAPPPNPP"                              <056>
60220 PRINT"KPLSFSPPRPSPPDPSSPPAASPLPPPPPP
      PPPPPNPPP"                              <146>
60222 PRINT"KPLSFSPPRPSPPDPSSPPPPPPLPPPPPP
      PPPPNPPPP"                              <165>
60224 PRINT"KPLSFSPPPPSPPDPSSAAPPPPLPPPPPP
      PPPNPPPPP"                              <033>
60226 PRINT"KPLSFSP5PPPPPDPSPP6PPPPLPPPPPP
      PPNPPPPPP"                              <144>
60228 PRINT"KPLSFSPPIILPPDPSPPAAPPPLPPPPPP
      PNPPPPPPP"                              <109>
60230 PRINT"KPLSFSPPPPLPPDPSPPPAAPPLPPPPPP
      NPPPPPPPP"                              <126>
```

```
60234 PRINT"KPLSFSPPPPLPPDPSPPPPPPPLPPPPPN
       PPSPPPPPP"                       <077>
60237 PRINT"KPRSFSPPPPLPPDPSAAAPPPPLPPP4NP
       PPSPPPPRP"                       <034>
60240 PRINT"KPPSFSPPPPLPPDPSPPPAAAPLPPPNPP
       PPSPPPAAP"                       <117>
60244 PRINT"KPPSFSPPPPLPPDPSPPPPPKPLPPRPPP
       PPSAAAPPP"                       <164>
60247 PRINT"KPPSFSPPPPIIIDIIIIIIIKPLPPMPPP
       PPPPPPPPP"                       <013>
60250 PRINT"KPTSLSPPPPPPPDPPPPPPPPPLPPPMPP
       PPPPPPAAP"                       <115>
60254 PRINT"KPPSLSPPPPPPPDPPPPPPPPPLPPPPMP
       PPSPPP2PP"                       <020>
60258 PRINT"KJJJJSPPRPP1PDPPPPPPPPPLPPPPPM
       PPSAAAPPP"                       <220>
60260 PRINT"PPPPPPPPAAAPPDPPPPPPPPPLPPPPPP
       MPPPPPPPP"                       <086>
60263 PRINT"PPPPPPPPPPPPPPDPPPPPPPPPLPPPPPP
       PMPPPPAAA"                       <123>
60264 PRINT"PPPPPPPPPPPPPPDPPPPRPP3PLPPPPPP
       PPMPPPPPP"                       <210>
60270 PRINT"AAAAAAAAAAAAABAAAAAAAAAAAAAAAA
       AAAAAAAAA"                       <181>
60280 RETURN                            <137>
60299 REM BILD4                         <250>
60300 :                                 <075>
60301 PRINT"PAAARAAPPPPPPPPPPPPPPPPPPPPPPPS
       SSPRJJJJJ"                       <143>
60302 PRINT"PPPPPPPIIIIIIIRIIIIICIIIIIIILS
       SNJJ2PPPK"                       <223>
60303 PRINT"PPPPPPPPPPPPPPPRSSSSSLSEPPPPPLS
       NPPPPPPPK"                       <009>
60304 PRINT"PPPIIIIPPPPJJJJSSPPPLSEPPPPPLN
       PPPPPPPRK"                       <129>
60305 PRINT"PPPPPPPPPPPP3PPSPPPPLSEPPJAABI
       PPPIIPPAB"                       <237>
60306 PRINT"JJJPPPPPPPPPPPPPSARSSLSERPPPPPP
       PPPPPPPPP"                       <073>
60307 PRINT"PPPPPPPPPPPPPPPPSSMSSLSEIPPPPPR
       PPPPPPPPR"                       <050>
60308 PRINT"PPPPRJJJPPPPPPPPSSSPSLSEPRPPIII
       PPPIIPPIC"                       <176>
60309 PRINT"PPFHHHHHPPJJJPPPPSRSLSEPIPPPPP
       HHHHHHHHD"                       <110>
60310 PRINT"PPFSPPPPPPPPPPPPPPSPSLSEHHHHHHH
       PPPPPPPPD"                       <008>
60311 PRINT"PPFSPPPPPPFPPJJJJPSRSLSEPPPPPPP
       PPPPPPPPR"                       <005>
60312 PRINT"PRFSPPPPPPPPPPPPPPSPSLSEPPRPPPR
       PPPPPPPPJ"                       <134>
60313 PRINT"FAHSPPPPPPPPPPPIIPSRSLSEPPAPPJJ
       JPPOPPPAP"                       <250>
60314 PRINT"FPPPPPIIIPIIIPPPPSPSPSPPAHHHHH
       HHHHHHPPR"                       <171>
60315 PRINT"FPPPIIIPPPPPPPPPPPPPPPPPPSSSSSS
       SSSSSSAAA"                       <252>
60316 PRINT"IIIIIGGGGGGGGGGGGGGGGGGGGGGLSSS
       SSSSSSSSS"                       <040>
60317 PRINT"SDSSSSSSSSSSSSSSSSSSSSSSSSSLSSS
       PPPPPPPPS"                       <167>
60318 PRINT"SDPSSSSSSSSSSSSSSSSSSSSSSSSLSSS
       PPPRPPPRS"                       <044>
60319 PRINT"SDRSSSSSSSSSSSSSSSSSSSSSSSS0SSS
       RPPIPPPAS"                       <099>
60320 PRINT"SDPPPPPPPPPPPPPPPPPPPPPPPPPPPSSS
       IPPPPPPPS"                       <149>
60321 PRINT"SDP1PPPPPPPPPPPP4PPPPPPPPPPPPPPP
       PPPP5PPPS"                       <215>
60322 PRINT"ABAAAAAAAAAAAAAJJJJJJJJJJJJAAAA
       AAAAAAAAS"                       <107>
60380 RETURN                            <237>
60399 REM BILD5                         <222>
60400 :                                 <175>
60401 PRINT"AAARAQJPP1ARAGCHARAPPPPRJJJAAA
       PCPPPPPPP"                       <157>
60402 PRINT"PPPPSSSPPPPPPPDPPPPPPPPPPPPPPPP
       PKPPPPPPP"                       <124>
60403 PRINT"PPPPSSSPPPPPPPDPPPPPPPPPPPPPPPP
       PKJJPPPPP"                       <091>
60404 PRINT"PPPNSSSPPPPRPPDPPRPPPPPPMPPPPPP
       PPPPPPPPP"                       <206>
60405 PRINT"PPNPSSSIIIIIPPDPPPPPPPPPPPPPPPPP
       PPPPAAPPP"                       <010>
60406 PRINT"PPMPSSSPPPPPQPEPPPPPPPPPPPPPPPRP
       PPHHPPPPP"                       <108>
```

```
60407 PRINT"PPPMPSSCAPPPPPPEPPJJJCOPPOPPPOP
       POPPPPPPP"                       <096>
60408 PRINT"PPPIMPPDPSPPPPPEP3PPPDPPPPPPPPPP
       PPPPAAAPP"                       <215>
60409 PRINT"RPPPPMPDPSAARABAAPPPDPPPIIIIPJ
       JJJPPPPRP"                       <130>
60410 PRINT"AAPPPPMDPSPPPPPPPPPPPPDPP2PJJJJP
       PPPPPPAAP"                       <113>
60411 PRINT"QJJJPPPDPSAAAAAQJJJJDPPPPPPPPP
       PPPPIIIPP"                       <229>
60412 PRINT"PPPPPPPDPSPPPPPPPPPPPDPPPIIIIPP
       PIOIQPPPP"                       <059>
60413 PRINT"PQJJPPPDPSPPPPPPPPPPPDPPPPPPPPPP
       PKSSSSSCP"                       <235>
60414 PRINT"PPPPPPPPDPPPPPPPP4PPPPDPPPPPPPPP
       PKSSSSSLP"                       <033>
60415 PRINT"PPIIIPPBAAAAAAAAAAAAABAAAAAAAAAA
       AKSSSSSPP"                       <052>
60416 PRINT"PPPPPPPSSSSSSSSSSSSPQPPKPPLPPK
       PPLPPPSPS"                       <209>
60417 PRINT"IIPPPPPSSSSSSSSPPPPPDPPKPPLPPK
       PPLAACSPS"                       <222>
60418 PRINT"PPPPPPPSSSSSSSSPHHHPDPPKPPLPPK
       PPLPPDSLS"                       <204>
60419 PRINT"PPQJJPPPPPPPPPSSRSSSPDPPKPPLPPK
       PPLPDDSRS"                       <005>
60420 PRINT"PPPPPPPPPPPPPPPSSPSSSPDPPKPPLPPK
       PPLPDRSPS"                       <194>
60421 PRINT"IIPPPPPPPPPPPP0PPSSSPDPPKPPLPPK
       PPLPDPPLS"                       <211>
60422 PRINT"AAAAAAAAAQJJJJJAAAAIBIQQQQQQQQ
       QQQABAAHS"                       <192>
60480 RETURN                            <083>
60499 REM BILD6                         <197>
60500 :                                 <021>
60501 PRINT"PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
       PPPPPPPPQ"                       <040>
60502 PRINT"PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
       PPPQJJJPK"                       <166>
60503 PRINT"PPPPPPPPHHHQJPPPPPPPPPPPPPPPPPP
       PPPQPPPPK"                       <040>
60504 PRINT"PPPPPPPRPPPPPPPPJJJPPPPPPPPPPPPP
       PPPQPPPPK"                       <241>
60505 PRINT"PPPP4PPPPPPPPPPPPPPPPPPPPPPPPPPJ
       JJJQQJJJQ"                       <134>
60506 PRINT"PPPPPPPCPPPPPPPPPPPOJJPPPPPPPPRP
       PPPPPPPPP"                       <121>
60507 PRINT"PPPPPPNSMPPPPPPPPPPPPPPPPJJJQQJP
       PPPPPIIIQ"                       <220>
60508 PRINT"FPPPPNSSSMPPPPPPPPPPPPPPPPPPPPPP
       PPPPPPPPP"                       <121>
60509 PRINT"FPPPNSSPSSMPPPPPPPPPPPPPPPPPPPPP
       PPPPJJPPP"                       <229>
60510 PRINT"FPPNSSPRPSSMQQQPQQQPQQQQQMPPPP
       PPPPPPPPP"                       <119>
60511 PRINT"FPNSSSPTPSSSMQQRQQQRQQQQQQPPPP
       PPPPIIIIQ"                       <195>
60512 PRINT"FQSSSSSSSSSSSSIIIIIIIIIIIIIMPPP
       PPPPPSPPK"                       <231>
60513 PRINT"RPSPRPSSSPPPSSSSSSSSSSSSSSSMPP
       PPPPPQPPK"                       <203>
60514 PRINT"IRSPTPSSSPRPSSSPPPSSSSPPPSSSMP
       PPPPPLPPK"                       <127>
60515 PRINT"PRSSDSSSSSNSSSS2RPSSSSPR3SSSSM
       PPPPPKPPK"                       <049>
60516 PRINT"5HSSDSSSSNSSSSSSTSSSSSSTSSSSSS
       MPPPPKPPS"                       <167>
60517 PRINT"GRSSDSSPPPPSSSSSSSSSSSSSSSSSSSS
       SPPPPKPPQ"                       <061>
60518 PRINT"RHSSDSSGGGRSPPPPPPPPPPPPPSPPPPS
       SRPPPKPPK"                       <092>
60519 PRINT"GRSSDSSPPPTSRPRRPPPRRPPRSPAARS
       SIIIIKPPK"                       <240>
60520 PRINT"RHSSDSSTTTTSIIIIQIIIIQIIMPPPRS
       SIPSSSSPK"                       <097>
60521 PRINT"RPP1DPPPPPPSSSSSSSSSSSSSSSMPRRS
       SPPSSSS0K"                       <076>
60522 PRINT"AAAAAAAAAAAASSSSSSSSSSSSSSSAQAAI
       IIQSSSAAB"                       <047>
60580 RETURN                            <183>
```


Listing »Springvogel« (Schluß)





Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Chefredakteur: Michael Scharfenberger
Leitender Redakteur: Albert Absmeier
Redaktion: Volker Everts, Achim Hübner, Georg Klinge, Harald Meyer, Markus Ohnesorg, Thomas Röder, Arnd Wängler.
Praktikant: Boris Schneider
Fremdautoren:
Behrend, Biernaczyk, Bölcskei, Büttgenbach, Burgmeister, Carli, Haas, Haberl, Hinkelmann, Illetschko, Kloeker, Kulzer, Lonczewski, Mann, Menke, Paetzold, Pfost, Rausch, Reiter, Rickert, Schäfer, Schindowski, Schoeberl, Schramm, Spitzner, Toerk, Trierscheid, Weineck, Weißbecker, Ziesche

Layout: Leo Eder (Ltg.)

Herstellung: Klaus Buck

Auslandsrepräsentation:
Schweiz: Markt & Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042-223155, Telex: 862329
USA: M&T Publishing Inc.; 2464 Embarcadero Way, Palo Alto, CA 94303

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten werden, so muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlags AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt & Technik Verlag AG verlegten Publikationen und dazu, daß Markt & Technik Verlag AG Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Vertriebsleitung: Hans Hörl

Anzeigenverwaltung und Disposition: Michaela Hörl

Verlagsleiter M&T-Buchverlag: Günther Frank

Druck: R. Oldenbourg GmbH, Hürderstraße 4, 8011 Kirchheim.

Preis: Das Einzelheft kostet DM 14,—

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebs GmbH, Hauptstätter Straße 96, 7000 Stuttgart 1, Telefon (0711) 764830

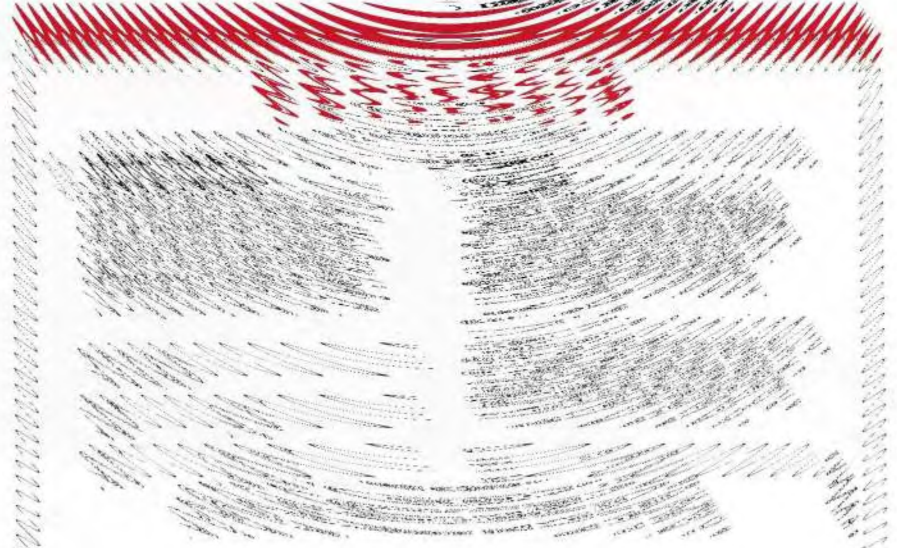
Verantwortlich:
Für redaktionellen Teil: Michael Scharfenberger
Für Anzeigen: Brigitte Fiebig
Redaktions-Direktor: Michael M. Pauly

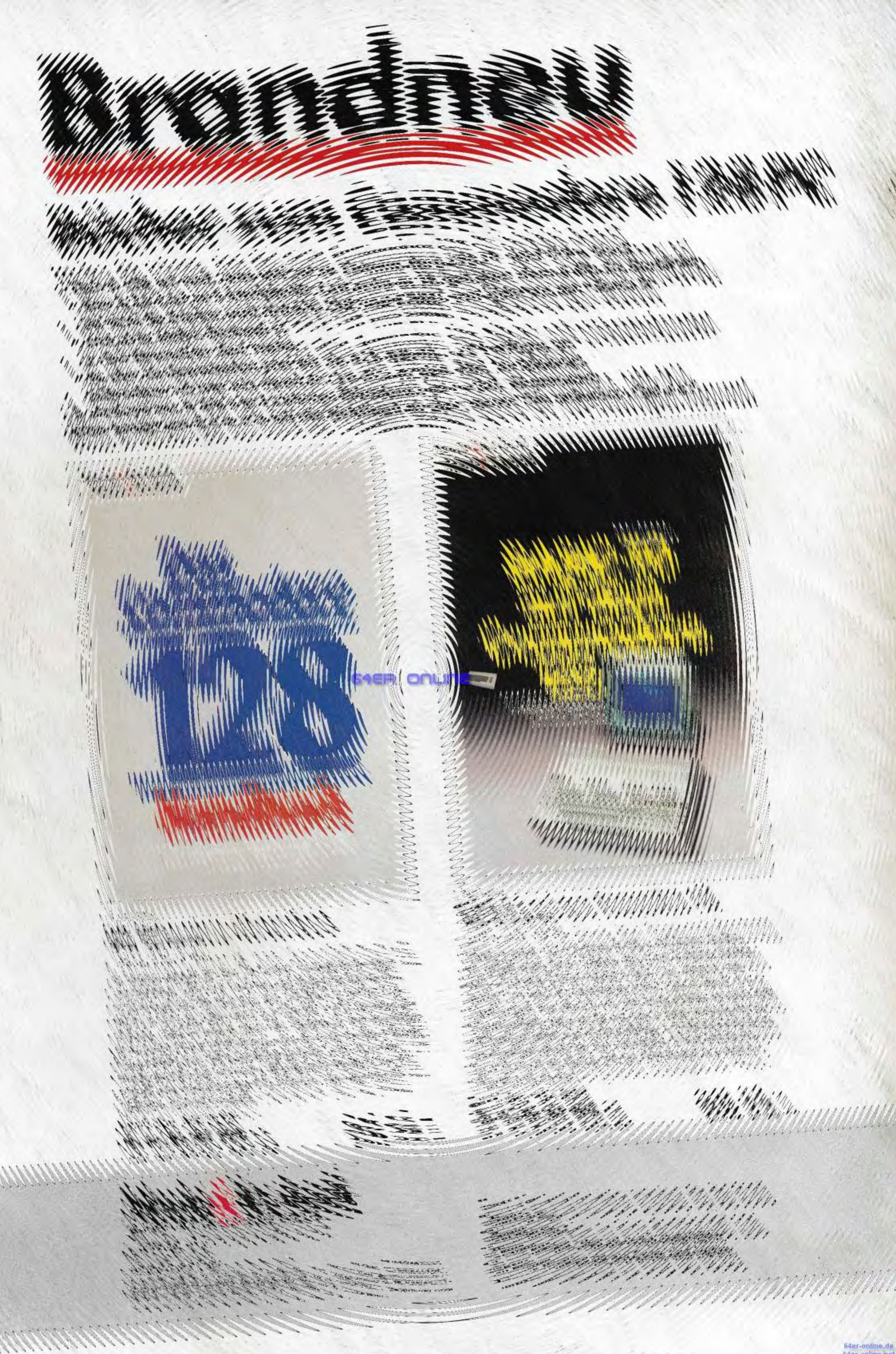
Vorstand: Carl-Franz von Quadt, Otmar Weber

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0, Telex 5-22052

Aktionäre, die mehr als 25% des Kapitals halten: Otmar Weber, Ingenieur, München; Carl-Franz von Quadt, Betriebswirt, München. Aufsichtsrat: Dr. Robert Dissmann (Vorsitzender), Karl-Heinz Fanselow, Eduard Heilmayr






128